Using Case-Based Reasoning and Argumentation to Assist Medical Coding

Michael Schnell

UNIVERSITÉ DE LORRAINE

École Doctorale IAEM Lorraine

Using Case-Based Reasoning and Argumentation to Assist Medical Coding

THÈSE

présentée et soutenue le 30 septembre 2020

pour l'obtention du

Doctorat de l'Université de Lorraine

(spécialité informatique)

par

Michaël Schnell

Composition du jury

Rapporteurs :	Mirjam Minor	*Professeur, Goethe-Universität*
	Catherine Duclos	*Professeur, Université Paris 13*
Examinateurs :	Sylvie Després	*Professeur, Université Paris 13*
	Horatiu Cirstea	*Professeur, Université de Lorraine*
	Julien Henriet	*Maître de conférences, Université de Franche-Comté*
Encadrants :	Jean Lieber	*Maître de conférences, Université de Lorraine*
	Nicolas Jay	*Professeur, Université de Lorraine*
Invitée :	Sophie Couffignal	*Luxembourg Institute of Health*

Laboratoire Lorrain de Recherche en Informatique et ses Applications

Acknowledgements

A thesis is a long and solitary process, yet gone are the times they are achieved alone. Behind every successful PhD, there is a team providing invaluable support, both human and professional needed to overcome this task. During my PhD, I have the luck to have such a team, helping me succeed where I would have failed otherwise. I am grateful to these people, even if it does not always show in my behaviour.

My colleague and friend David Marcic was my supervisor during my master internship at Luxembourg Institute of Health (LIH). He encouraged me to pursue a PhD and helped me find my subject. To this day, his counsel is invaluable and he plays an important part in my success so far. He has supported me greatly with his jokes, attitude and insights, and I am immensely grateful to him. I hope we will continue to work and flourish together in our new missions at LIH.

My dear friend and fellow PhD candidate Tatjana Makovski provided me the much needed support. Thank you for pushing me to do my best and never accepting my weak excuses for postponing.

Sophie Couffignal is the operational head of the Luxembourg National Cancer Registry (NCR). Her patience and support, enables us to perform our research. I would also like to thank Dr Michel Untereiner, scientific head of the NCR, who trusted me and helped us obtain our initial funding for the thesis from the Fondation Cancer. Thank you to our head of department, Laetitia Huiart, for her support during the last years of my PhD and for her trust in me.

I am very grateful to both my supervisors Nicolas Jay and Jean Lieber, without them this PhD would not have been possible. They were very patient with me and guided me with their expertise. They made themselves available when I needed them and helped me stay on track. Their advice both for technical aspects and general research practices were very insightful. I am glad to have had the opportunity of doing my PhD with them.

I would also like to thank both my teams at LIH (IT-X-HD unit) and at Loria (*équipe K*) for their advice and the fun times spent working together.

Abstract

The aim of the National Cancer Registry (NCR) in Luxembourg is to collect data about cancer and the quality of cancer treatment. To obtain high quality data that can be compared with other registries or countries, the NCR follows international coding standards and rules, such as the International Classification of Diseases for Oncology (ICD-O). These standards are extensive and complex, which complicates the data collection process. The operators, i.e. the people in charge of this process, are often confronted with situations where data is missing or contradictory, preventing the application of the provided guidelines. To assist in their effort, the coding experts of the NCR answer coding questions asked by operators. This assistance.is time consuming for experts. To help reduce this burden on experts and to facilitate the operators' task, this project aims at implementing a coding assistant that would answer coding questions. From a scientific point of view, this thesis tackles the problem of extracting the information from a set of data sources under a given set of rules and guidelines. Case-based reasoning has been chosen as the method for solving this problem given its similarity with the reasoning process of the coding experts. The method designed to solve this problem relies on arguments provided by coding experts in the context of previously solved problems. This document presents how these arguments are used to identify similar problems and to explain the computed solution to both operators and coding experts. A preliminary evaluation has assessed the designed method and has highlighted key areas to improve. While this work focused on cancer registries and medical coding, this method could be generalized to other domains.

Résumé

Le but du Registre National du Cancer (RNC) du Luxembourg est de collecter des données sur le cancer et la qualité des traitements au Luxembourg. Afin d'obtenir des données de haute qualité et comparables avec celles d'autres registres ou pays, le RNC suit les règles et standards internationaux de codification comme la Classification International des Maladies pour l'Oncologie (COM-O). Ces standards sont complexes et considérables, compliquant fortement le processus de collecte des données. Les encodeurs en charge de la collecte des données sont souvent confrontés à des situations dans lesquelles des données sont manquantes ou contradictoires, les empêchant d'appliquer les règles fournies. Pour les aider dans leur tâche, les exports de codification du RNC répondent aux questions de codage des encodeurs. Cependant, ces réponses requièrent beaucoup de temps des experts. Le but de ce projet est de réduire le temps d'expert nécessaire et de faciliter le travail des encodeurs. D'un point de vue scientifique, cette thèse s'intéresse au problème de synthèse d'informations à partir d'un ensemble de données provenant de différentes sources avec des contraintes et recommandations à respecter. Le raisonnement à partir de cas est utilisé pour résoudre ce problème car cette méthodologie ressemble à cette employée par les experts. La méthode de résolution conçue utilise des arguments fournis par les experts de codification dans le cadre de questions posées précédemment par les encodeurs. Ce document décrit comment ces arguments servent à identifier des questions similaires et à expliquer la réponse aux encodeurs et aux experts. Une évaluation préliminaire a été réalisée pour évaluer la performance de la méthode et identifier des pistes d'améliorations. Dans un premier temps, le travail produit porte sur les registres du cancers et la codification médicale, cependant l'approche est généralisable à d'autres domaines.

Document Structure

This document details the work accomplished in the context of my doctoral studies. This document is composed of two parts. The first chapter is a French summary of the whole thesis, as requested by the doctoral school. The remainder of the document is the thesis itself, written in English.

Contents

Chapter 1

Assistance au codage médical par du raisonnement à partir de cas argumentatif

Application aux registres du cancer

Résumé français

1.1 Introduction

Dans les pays développés, le cancer est l'une des principales causes de décès [World Health Organization, 2019]. Afin de diminuer l'impact de cette maladie sur la société, plusieurs mesures sont mises en place. Pour identifier les mesures les plus appropriées et pour évaluer leur impact, des données sur la situation du cancer sont nécessaires. Pour cela, des registres du cancer peuvent être utilisés.

1.1.1 Registres du Cancer

Afin de pouvoir planifier la lutte contre le cancer, il faut avoir une vision claire de la situation, notamment de la prévalence, l'incidence et la prise en charge des patients atteints de la maladie. Pour cela, de nombreux pays ont mis en place des registres du cancer. En 2013, le Luxembourg s'est également doté d'un tel registre avec la création du Registre National du Cancer (RNC). Le RNC est une base de données exhaustive et non redondante des cas de cancer diagnostiqués et/ou traités au Luxembourg. Dans ce registre sont recensés les types de cancer et la façon dont ils ont été diagnostiqués et traités. Ce registre fournit les données nécessaires pour l'évaluation des mesures de santé publique au Luxembourg et de la situation du cancer.

Afin de pouvoir comparer la situation au Luxembourg avec d'autres pays, il est important que la collecte des données et la codification suivent des standards communs. Pour les registres du cancer, il existe plusieurs standards, couvrant différents aspects de la codification, comme la Classification Internationale des Maladies pour l'Oncologie (CIM-O) [World Health Organisation, 2013].

1.1.2 Codification

Les standards de codification doivent couvrir un grand nombre de situations, car le cancer est une maladie très diverse et il y a beaucoup d'éléments à prendre en compte. Malgré cette complexité, les standards ne parviennent pas à couvrir tous les cas imaginables, ce qui complique le travail des encodeurs. Afin de les assister et pour pallier à l'absence de consignes claires, des standards et des recommandations ont été élaborées, comme celles de l'*European Network of Cancer Registries* (ENCR) [Tyczynski

et al., 2003]. Chacun de ces registres dispose également de ses propres recommandations adaptée à son contexte local.

Pour avoir des données de haute qualité, il est important d'appliquer correctement et de façon cohérente toutes ces règles et recommandations. Pour cela, il faut former longuement les encodeurs et les encadrer continuellement, pour s'assurer qu'ils/elles connaissent les règles et consignes et leurs évolutions. En plus, les situations rencontrées par les encodeurs peuvent aussi être difficiles à interpréter. Dans les dossiers patients, il est possible de retrouver des informations vagues, contradictoires ou encore manquantes. Cela complique l'application des standards. Pour aider les encodeurs dans leur tâche, le RNC a mis en place un système permettant aux encodeurs de poser des questions aux experts de codification du registre. Dans leurs questions, les encodeurs décrivent sous forme de texte les éléments du dossier patient qu'ils/elles jugent pertinents et formulent leur questionnement.

Avec ces informations anonymisées, les experts de codifications du RNC tentent de répondre au mieux aux questions. Ils doivent respecter les standards suivis, mais aussi les éventuelles précédentes décisions de codification. Ce dernier point est notamment important pour assurer la cohérence des données. Cependant, comme les questions et les réponses ne sont pas structurées, il n'y a pas de moyen fiable pour les experts et les encodeurs pour retrouver les situations similaires. Les réponses fournies par les experts sont discutées avec les encodeurs lors d'ateliers mensuels de codification organisés par le RNC.

Ce projet a été initié afin d'aider les experts et les encodeurs dans leur travail pour le registre. A première vue, l'approche des experts ressemble au raisonnement à partir de cas, qui est une méthode de résolution de problèmes utilisant des anciens problèmes résolus. Cette méthode peut être utilisée avec des technologies du web sémantique, notamment pour la représentation des connaissances.

Ce chapitre résume les travaux réalisés dans le cadre de ce projet, en commençant par introduire les notions pertinentes pour la méthode conçue, suivi d'une description de la représentation choisie, de la méthode conçue et de l'évaluation préliminaire réalisée.

L'objectif scientifique est d'analyser comment les informations provenant de différentes sources peuvent être fusionnées en respectant des contraintes, tout en fournissant une explication de la solution. L'objectif applicatif est de réaliser un assistant de codification utilisant la méthode conçue pour le RNC dans premier temps.

1.2 Préliminaires

L'une des principales difficultés concerne l'identification des règles ou recommandations à appliquer. Dans l'approche actuelle en place au RNC, les informations sont enregistrées dans le système de tickets utilisé, les messages électroniques échangés avec les encodeurs et les comptes-rendus des ateliers. Lorsqu'un expert doit répondre à une nouvelle question, il doit se fier à sa mémoire pour déterminer rapidement s'il y a une autre question similaire qui a déjà été posée. Vu le temps limité des experts de codification, ces derniers ne peuvent pas se permettre de revoir toutes les anciennes questions. Or, pour garantir la qualité des données du registre, il est important que les décisions de codification des experts respectent les choix précédents. Ainsi, si deux questions portent sur des situations similaires, souvent les réponses sont similaires.

L'énorme quantité de règles et recommandations pose également un grand défi pour les encodeurs. Comme les experts, les encodeurs ne disposent pas de moyen efficace pour retrouver la bonne manière de codifier un cas complexe.

Cette section introduit les notions et méthodes utilisées pour assister les membres d'un registre du cancer dans leur travail de codification des données.

1.2.1 L'intelligence artificielle explicable

Une explication de la réponse fournie par un système de résolution de problème est un critère de qualité important [Gregor and Benbasat, 1999]. Récemment, un nouveau domaine, appelé intelligence artificielle explicable ou XAI (*explainable artificial intelligence*), a émergé en intelligence artificielle ayant pour but la conception et l'analyse d'algorithmes explicables [Van Lent et al., 2004]. En expliquant la solution fournie, il est plus facile pour un utilisateur d'accepter une solution, de la critiquer ou de l'améliorer. Cela permet de faciliter l'acceptation d'un système par les utilisateurs. Ces explications sont également de plus en plus nécessaires d'un point de vue légal. Avec la multiplication des traitements automatiques de demandes, une personne doit pouvoir comprendre la suite donnée à sa demande, en particulier en cas de refus. Ce ¡¡ droit à l'explication ¿¿ est notamment présent dans la loi française [Bygrave, 2001] et la Réglementation Générale de Protection des Données dans l'espace économique européen.

Pour rendre un système explicable, une approche consiste à y associer un deuxième système qui fournit une explication compréhensible par un utilisateur pour la réponse fournie [Nugent et al., 2009, Olsson et al., 2014]. Il y a également des approches qui sont ¡¡ naturellement ¿¿ explicables, au sens où un utilisateur peut suivre les grandes étapes de résolution d'un problème. C'est en général le cas pour le raisonnement à partir de cas.

Afin de structurer la description d'un problème, il faut définir un langage de représentation de connaissances. RDFS (*Resource Description Framework Schema*) est un tel langage [Brickley and Guha, 2014], utilisé dans le cadre du web sémantique.

1.2.2 RDFS et SPARQL

RDFS est un langage générique de représentation de données et de connaissances. En RDFS, les données sont représentées par des triplets (`suj pred obj`), qui peuvent être vus comme des phrases, où `suj` représente le sujet, `pred` (prédicat) un groupe verbal et `obj` l'objet. Une base RDFS est un ensemble de triplets et peut être assimilée à un graphe. Par exemple, la base RDFS suivante contenant trois triplets

(`rachmaninov né_en 1873`)
(`rachmaninov a_composé danses_symphoniques`)
(`danses_symphoniques a Symphonie`)

peut être assimilée au graphe

$$\texttt{1873} \xleftarrow{\texttt{né\_en}} \texttt{rachmaninov} \xrightarrow{\texttt{a\_composé}} \texttt{danses\_symphoniques} \xrightarrow{\texttt{a}} \texttt{Symphonie}$$

et indique que Rachmaninov est né en 1873 et a composé la symphonie ¡¡ Danses symphoniques ¿¿.

Tous les éléments utilisés dans les triplets RDFS sont appelés ressources. Ces ressources sont de types différents, des URI (*Universal Resource Identifier*) ou des littéraux typés.

Contrairement aux systèmes de bases de données classiques (comme les bases de données relationnelles), il est possible d'inférer des connaissances qui ne sont pas explicitement représentées dans une base RDFS. Pour cela, certaines ressources sont associées à une sémantique permettant à un moteur d'inférence de générer des nouvelles connaissances. C'est le cas par exemple pour les ressources décrivant des classes d'éléments ou encore pour les propriétés `rdf:type` (souvent abrégé par `a`) et `rdfs:subClassOf` (souvent abrégé par `subc`). Une *classe* est un type de ressource permettant de regrouper des éléments similaires dans un ensemble, comme la classe des animaux. Une *instance* est un élément particulier d'une classe, comme Rex le chien qui est une instance de la classe des animaux. Pour indiquer qu'une ressource appartient à une classe, la propriété `a` est utilisée. Ainsi le triplet (`rex a Animal`) indique que Rex appartient à la classe des animaux. Il est possible de définir des hiérarchies de classes grâce à la propriété `subc`. Ainsi le triplet (`Chien subc Animal`) indique que la classe des chiens est une sous-classe de la classe des animaux.

SPARQL Protocol and RDF Query Language (SPARQL) [Harris and Seaborne, 2013] est un langage créé pour manipuler et interroger en particulier des bases RDFS. La majorité des actions sont décrites à l'aide de requêtes. Une requête est composée d'un type (SELECT pour récupérer des informations, INSERT pour ajouter des informations, etc.) et d'un ensemble de contraintes sur un graphe RDFS. Ces contraintes permettent d'identifier des sous-graphes partiels RDFS, correspondant aux triplets à extraire de la base de connaissances interrogée.

Dans le contexte de ce projet, les requêtes principales sont de type `ASK`. Ces requêtes permettent d'interroger une base de connaissances afin de déterminer s'il existe un sous-graphe partiel correspondant aux contraintes indiquées dans la requête. Par exemple, La requête

```
ASK {
  ?a a_composé ?piece .
  ?piece a Symphonie
}
```

teste l'existence d'un sous-graphe partiel RDFS représentant un auteur d'une symphonie. L'élément `?a` représente une variable dans le langage SPARQL (noms commençant par un `?`). Cette requête retourne la valeur `VRAI` pour la base RDFS introduite au début de cette section.

1.2.3 Les distances d'édition

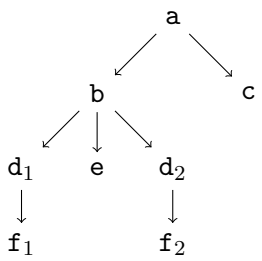
Il y a plusieurs approches pour déterminer la différence ou la distance entre deux objets. Pour calculer la distance entre deux chaînes de caractères, la distance de Levenshtein [Levenshtein, 1966] peut être utilisée. Cette distance repose sur le coût d'opérations d'édition nécessaires pour transformer un objet en l'objet comparé. Pour cette distance, ces opérations sont l'insertion d'un caractère, la suppression d'un caractère et la substitution d'un caractère par un autre. D'autres distances peuvent faire appel à d'autres opérations d'édition. Chaque édition a un coût et la distance d'édition est définie comme le coût minimal des opérations nécessaires pour transformer un object en un autre. Par exemple, en supposant que toutes les opérations ont un coût fixe de `1`, la distance entre les chaînes `train` et `avion` est `4` (supprimer `t` et `r` en début de chaîne et insérer un `v` après le `a` et un `o` après le `i`).

Il est possible de définir une distance d'édition pour les graphes [Bunke and Messmer, 1993]. Comme pour la distance de Levenshtein, trois opérations d'édition sont utilisées (insertion, suppression et substitution). Afin de limiter le nombre d'opérations à considérer pour le calcul de la distance, il est possible de limiter les opérations aux nœuds du graphe. Pour les arbres, qui sont des cas particuliers de graphes, il est possible de réduire davantage les opérations d'édition à considérer en utilisant une approche par niveau. Pour calculer la distance d'édition entre deux arbres `source` et `cible`, la première opération consiste à substituer la racine de l'arbre `source` par celle de l'arbre `cible`. Ensuite, les fils directs des deux racines sont pris en compte pour le calcul de la distance, avant de continuer avec les fils de ces nœuds, jusqu'à l'obtention de l'arbre `cible`. La figure 1.1 illustre la distance d'édition entre deux arbres. En supposant que le coût des opérations est fixé à `1`, la distance entre ces deux arbres est de `8`.

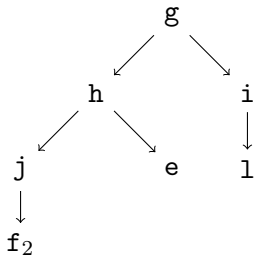
1.2.4 Le raisonnement à partir de cas

Le raisonnement à partir de cas [Aamodt and Plaza, 1994] est une méthode générique de résolution de problèmes, utilisant des problèmes résolus précédemment et leurs solutions pour résoudre de nouveaux problèmes. Cette méthode dérive de l'idée que des problèmes similaires ont souvent des solutions similaires. Ainsi, pour trouver la solution d'un problème, un problème similaire et sa solution peuvent être utilisés.

Un *cas* est la représentation d'un épisode de résolution d'un problème. Un cas est typiquement représenté par un couple (`pb`, `sol(pb)`), où `pb` est un problème du domaine d'application et `sol(pb)` est la solution retenue pour ce problème. Étant donné un nouveau problème `cible`, aussi appelé problème cible, le raisonnement à partir de cas a pour but de résoudre ce problème en utilisant un cas source. Un *cas source* est un cas de la *base de cas*, c'est-à-dire l'ensemble des problèmes résolus



(a) Arbre source.



(b) Arbre cible.

- substituer `a` par `g` ;
- supprimer `c` ;
- insérer `i` ;
- insérer `l` ;
- substituer `b` par `h` ;
- substituer d_2 par `j` ;
- supprimer d_1 ;
- supprimer f_1.

(c) Opérations d'édition.

Figure 1.1: Deux arbres et un exemple d'opérations d'édition.

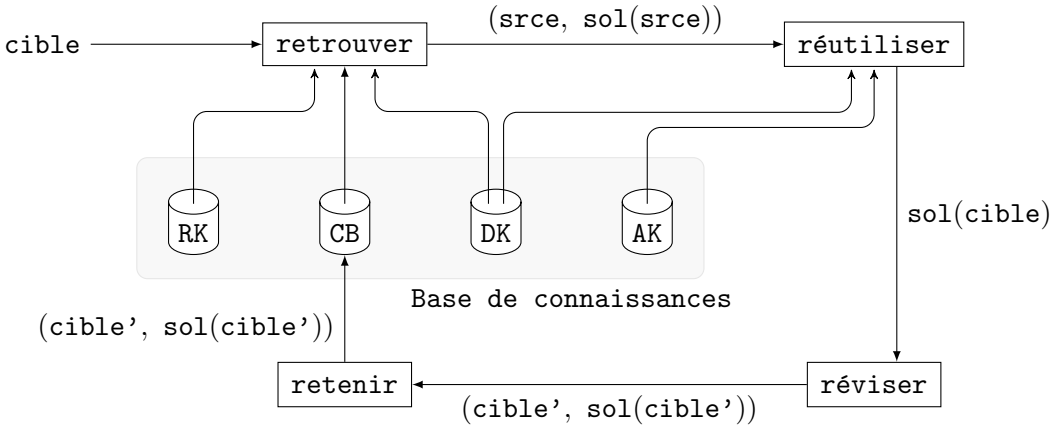


Figure 1.2: Raisonnement à partir de cas avec un cycle quatre R et quatre conteneurs de connaissances.

et leurs solutions. L'approche classique consiste à identifier le cas source le plus proche du problème cible et à l'utiliser pour déterminer la solution du problème cible. La définition exacte de la notion de *similarité* dépend du domaine d'application.

Le raisonnement à partir de cas est une méthode générale et peut être implémenté de différentes façons. Parmi les approches possibles, on retrouve le cycle *4-R* [Aamodt and Plaza, 1994], illustré dans la figure 1.2. Le cycle est découpé en quatre étapes, les quatre ¡¡ R ¿¿ : retrouver (*retrieve*), réutiliser (*reuse*), réviser (*revise*) et retenir (*retain*). L'étape réutiliser s'accompagne parfois d'une modification de la solution du cas source. On parle alors d'adaptation du cas source.

Le but de l'étape retrouver est d'identifier le cas source (`srce`, `sol`(`srce`)) le plus proche du problème cible `cible`, appelé cas remémoré. Ce cas est utilisé lors de l'étape réutiliser pour déterminer une solution `sol`(`cible`) pour le problème cible. Le nouveau cas (`cible`, `sol`(`cible`)) peut ensuite être révisé, par exemple manuellement par un expert du domaine, pour valider la solution fournie ou encore pour modifier la description du problème `cible`. Dans l'étape retenir, le nouveau cas (`cible'`, `sol`(`cible'`)) peut alors être ajouté dans la base de cas du système, permettant au système de potentiellement résoudre des problèmes supplémentaires.

Dans chacune des étapes du raisonnement à partir de cas, des connaissances du domaine sont nécessaires. Dans [Richter and Weber, 2013], ces connaissances sont regroupés dans quatre conteneurs de connaissances (*knowledge containers*). Ce découpage permet de décrire à quelle étape les connaissances sont utilisées, comme illustré dans la figure 1.2. Les quatre conteneurs sont les connaissances du domaine (DK – *domain knowledge*), les connaissances pour retrouver (RK – *retrieval knowledge*), la base de cas (CB – *case base*) et les connaissances d'adaptation (AK – *adaptation knowledge*).

1.2.5 L'argumentation

L'un des avantages de l'argumentation est l'explication. Avec l'explosion du nombre d'outils d'aide à la décision et l'obligation légale de justifier les décisions, beaucoup de travaux scientifiques ont porté sur l'utilisation d'arguments dans des systèmes d'intelligence artificielle. Précédemment, des travaux ont déjà démontré l'utilité des arguments dans un cadre légal. Les systèmes HYPO [Ashley, 1991] et CATO [Aleven and Ashley, 1997] sont deux exemples, ayant pour objectif d'aider des avocats à défendre ou critiquer une position grâce à des précédents juridiques. Ces systèmes identifient ces précédents et mettent en évidence les similarités entre le précédent et la situation actuelle.

Des combinaisons entre l'argumentation et le raisonnement à partir de cas ont également été étudiées [Karacapilidis et al., 1997, Ontañón et al., 2015].

1.3 Représentation des connaissances

Afin de pouvoir manipuler les questions des encodeurs et les réponses des experts, il faut définir la façon de les représenter.

1.3.1 Représentation des cas

Pour appliquer le raisonnement à partir de cas, il faut définir ce que représente un cas. Ce projet se déroule dans le cadre de l'aide au codage pour un registre du cancer. Ainsi, les problèmes à résoudre concernent des questions de codification posées par des encodeurs. Un cas est défini par un couple (`pb`, `sol(pb)`), ou `pb` est un problème et `sol(pb)` une solution de ce problème.

Un problème est composé d'une question et d'une description du dossier patient concerné. Une question est composée de plusieurs éléments, notamment le sujet de la question, le type de cancer concerné et la version des standards de codification à appliquer. La description du dossier patient contient une brève description du patient (âge et sexe) et une liste des examens pertinents par rapport à la question, ainsi que les observations et conclusions pour ces examens.

Une solution est composée de la réponse fournie à la question et d'une argumentation de cette réponse. Dans le cas d'une question portant sur la topographie d'une tumeur, cette réponse est un code topographique de la CIM-O. La topographie codifie le lieu de départ d'un cancer, c'est-à-dire l'endroit du corps où le cancer a débuté. L'argumentation est une liste d'arguments favorables et défavorables à la réponse fournie. Un argument représente une partie du raisonnement de l'expert de codification pour répondre à une question. Ces arguments sont utilisés pour l'identification et pour l'explication de la réponse.

Un argument peut être découpé en trois parties :

- les éléments pertinents du dossier patient,
- les connaissances du domaine (médical et de codification) intervenants dans le raisonnement *et*
- les réponses soutenues.

Par exemple, l'argument suivant peut être utilisé dans le contexte d'une question portant sur la topographie d'une tumeur pulmonaire :

> *Un adénocarcinome situé dans les poumons testant positif pour le marqueur TTF1 est un élément favorable à une tumeur primitive du poumon.*

Pour cet argument, les éléments pertinents du dossier patient sont :

- la présence d'une tumeur pulmonaire,
- l'identification de la morphologie de cette tumeur en tant qu'adénocarcinome *et*
- la présence d'un test positif sur la tumeur pour le marqueur TTF1.

Les connaissances médicales portent sur le lien entre la présence du marqueur TTF1 pour les adénocarcinomes et la nature primitive de ces tumeurs. La nature d'une tumeur (primitive ou secondaire) est définie par la localisation initiale où la tumeur s'est développée. Ainsi la tumeur primitive

est la tumeur initiale. Les métastases, c'est-à-dire les nouvelles tumeurs qui se sont développées en dehors du site d'origine, sont de nature secondaire.

Pour les réponses soutenues par cet argument, il s'agit des codes topographique du poumon, c'est-à-dire les codes `C34.0` à `C34.9`.

Considérons l'exemple d'une question portant sur la topographie d'une tumeur. Cette question concerne un patient diagnostiqué en 2016. Initialement, une imagerie met en évidence une tumeur au niveau du lobe inférieur droit du poumon. Une biopsie de cette tumeur identifie cette tumeur comme étant un adénocarcinome. Un test pour le marqueur TTF1 est négatif. Un *PET scan* du corps complet ne trouve pas de tumeur additionnelle. Ce patient est discuté en réunion de concertation pluridisciplinaire (RCP) et les cliniciens présents concluent pour une tumeur primitive du poumon. Un traitement chirurgical est suggéré. Le patient est opéré pour une résection de la tumeur. L'examen histologique de la pièce opératoire indique que la tumeur a été intégralement retirée.

Le poumon est une partie du corps dans laquelle se développent fréquemment des tumeurs métastatiques, c'est-à-dire des nouvelles tumeurs qui apparaissent en dehors du site d'origine. Ainsi, lorsqu'un encodeur est confronté à une tumeur pulmonaire, il faut déterminer s'il agit bien d'une tumeur primitive, car seules les tumeurs primitives sont à encoder dans le registre.

Pour la représentation d'un problème, une approche simple consiste à utiliser des couples attribut-valeur. Cependant, cette approche ne permet pas de représenter facilement des liens entre des éléments. Or, pour le domaine d'application de ce projet, il est important de représenter ces liens. En effet, une information n'a pas la même crédibilité en fonction de l'examen (source) qui l'a fournie. Par exemple, pour identifier la morphologie d'une tumeur, c'est-à-dire le type de cellules et le comportement de la tumeur, l'avis d'un anatomopathologiste compte plus que l'avis d'un radiologue.

RDFS est une alternative utilisée dans certaines applications du raisonnement à partir de cas et c'est l'option retenue pour ce projet. Ce langage permet notamment l'utilisation de nombreux outils libres et fiables pour la manipulation et le stockage des connaissances. Afin de faciliter la maintenance des concepts utilisés, des concepts présents dans des bases de connaissances communes sont utilisés. C'est notamment le cas pour les codes topographiques, les codes morphologiques et les éléments du corps humains, qui sont représentés par des concepts définis dans l'ontologie SNMIFRE [CISMEF, 2015], qui est une traduction française de la classification SNMI (*Systematized Nomenclature of Medicine International*). La figure 1.3 montre un extrait du graphe RDFS associé au dossier patient de l'exemple introduit dans la section 1.3.1.

1.3.2 Représentation des arguments

La représentation des arguments doit prendre en compte les différentes finalités d'utilisation des arguments. Les arguments sont utilisés pour expliquer la réponse aux encodeurs et pour identifier le cas source le plus proche.

L'explication de la réponse se fait à l'aide d'un texte représenté par une chaîne de caractères et fourni par les experts de codification. Cette explication n'est pas exploitable sous cette forme par la méthode conçue dans ce projet, une étape de formalisation par un ingénieur de connaissances est nécessaire. L'idée sous-jacente pour identifier le cas remémoré est que des réponses similaires s'appuient souvent un raisonnement similaire. Le raisonnement des experts de codification est représenté par les arguments, et donc il est important de pouvoir déterminer si un argument s'applique à un problème. Si les arguments s'appliquent, alors le raisonnement validé des experts s'applique et il est possible de le réutiliser pour fournir une réponse à un nouveau problème. Comme indiqué dans la section 1.3.1, un argument repose en partie sur des éléments du dossier patient. Pour la méthode décrite dans cet article, un argument *s'applique* à un problème si les éléments sur lesquels il repose sont présents dans le dossier patient associé au problème. Ainsi, pour vérifier si un argument s'applique, il suffit de contrôler la présence de ces éléments dans le dossier patient. Comme celui-ci est représenté par un graphe RDFS, vérifier l'applicabilité d'un argument revient à chercher un sous-graphe partiel dans le graphe RDFS du dossier. Ce test peut facilement être réalisé à l'aide d'une requête `ASK` en SPARQL,

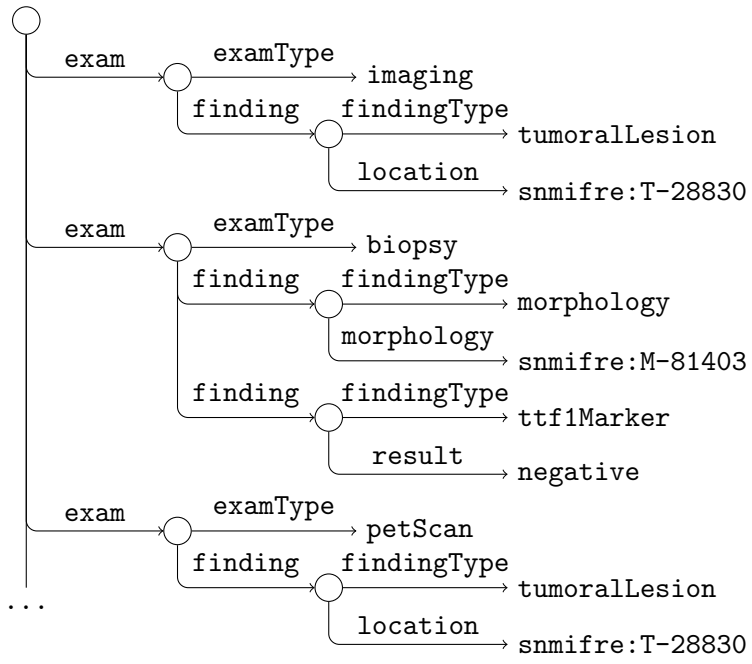


Figure 1.3: Extrait du graphe RDFS décrivant le dossier patient de l'exemple introduit dans la section 1.3.1. (`snmifre:T-28830` : lobe inférieur du poumon, `snmifre:M-81403` : adénocarcinome)

facilitant également la formalisation des arguments grâce à l'utilisation RDFS. La figure 1.4 montre la requête `ASK` associé à l'argument introduit dans la section 1.3.1.

Les deux autres parties des arguments ne sont pas représentées dans la version actuelle de la méthode. Néanmoins, elles pourraient être utiles pour une éventuelle adaptation des arguments et sont donc une piste intéressante pour des travaux futurs.

Pour la suite du document, étant donné un cas source $\texttt{srce}_i$, les arguments forts favorables associés à la solution de $\texttt{srce}_i$ sont dénotés par $\texttt{sp}_i^1$, $\texttt{sp}_i^2$, etc., les arguments faibles favorables par $\texttt{wp}_i^1$, $\texttt{wp}_i^2$, etc. et les arguments fiables défavorables par $\texttt{wc}_i^1$, $\texttt{wc}_i^2$, etc.

1.4 Raisonnement à partir de cas et argumentation

Cette section présente l'implémentation du raisonnement à partir de cas conçue pour répondre aux questions de codification posées par des encodeurs d'un registre du cancer. Cette approche se concentre sur l'utilisation d'arguments [Schnell et al., 2017], notamment lors des étapes retrouver et réutiliser.

1.4.1 Types d'arguments

Lors de l'analyse du raisonnement des experts de codification, trois types d'arguments ont été identifiés, les arguments forts favorables, les arguments faibles favorables et les arguments faibles défavorables. Le type d'un argument est défini par les experts de codification.

Un argument fort est un argument qui ne laisse aucun doute par rapport à la réponse à choisir. Un argument faible quant à lui ne permet pas d'affirmer la réponse avec certitude. Ce type d'argument est une indication qui encourage une réponse, sans pour autant exclure complètement une autre réponse.

Un argument est favorable s'il soutient la réponse à la question et il est défavorable s'il ne soutient pas la réponse. Ainsi un argument peut être favorable dans un cas et défavorable dans un autre cas. La force d'un argument ne dépend pas de la réponse choisie.

A noter qu'il n'y a pas d'argument fort défavorable. En effet, suivant la définition précédente, un tel argument serait un élément certain indiquant que la réponse choisie est fausse. Comme le but est de répondre à une question, il n'est pas pertinent de fournir une réponse dont il est connu qu'elle

```
ASK {
    # Lésion tumorale dans le poumon
    ?exam1 finding ?finding1 .
    ?finding1 findingType tumoralLesion ; location ?location .
    ?location subc snmifre:T-28000_S2 .
    # Adénocarcinome
    ?exam2 finding ?finding2 .
    ?finding2 findingType morphology ; morphology ?morph .
    ?morph subc snmifre:M-81400_S3 .
    # Marqueur TTF1
    ?exam3 finding ?finding3 .
    ?finding3 findingType ttf1Marker ; present yes .
}
```

Figure 1.4: Requête `ASK` associé à l'argument *Un adénocarcinome situé dans les poumons testant positif pour le marqueur TTF1 est un élément favorable à une tumeur primitive du poumon.*

est incorrecte. Cependant, il pourrait être utile d'étudier ce genre d'arguments pour permettre au système d'écarter des réponses lors de l'adaptation de la réponse.

Formellement, un argument est représenté par une fonction qui prend un problème et retourne un booléen (`VRAI` ou `FAUX`) indiquant si l'argument en question s'applique au problème.

Un cas est représenté par un couple (`pb`, `sol(pb)`), `pb` étant une description du problème et `sol(pb)` une solution pour ce problème. Les fonctions `sp`, `wp` et `wc` retournent les arguments de type fort et favorable, faible et favorable, et faible et défavorable.

Soit $\mathcal{N}_{\texttt{tgt}}^{\texttt{argt}}$ une fonction paramétrée par une fonction de type d'arguments $\texttt{argt} \in \{\texttt{sp}, \texttt{wp}, \texttt{wc}\}$ et un problème `cible` qui prend un cas source `srce` et retourne le nombre d'argument de `srce` du type d'argument concerné qui s'appliquent au problème `cible`. $\mathcal{N}_{\texttt{tgt}}^{\texttt{argt}}$ est défini par

$$\mathcal{N}_{\texttt{tgt}}^{\texttt{argt}}(\texttt{srce}) \;=\; |\{\texttt{arg} \in \texttt{argt}(\texttt{srce}) \mid \texttt{arg}(\texttt{cible}) \;=\; \texttt{VRAI}\}|$$

1.4.2 Exemple

Pour illustrer la méthode, le problème cible introduit dans la section 1.3.1 est résolu avec la base de cas contenant les trois cas sources suivants, tous portant sur la topographie de la tumeur.

Le premier cas source $\texttt{srce}_1$ concerne un patient, pour lequel un rapport d'imagerie met en évidence une lésion tumorale dans le poumon gauche. Le rapport d'une biopsie de cette tumeur indique qu'il s'agit d'un mélanome. Ce type de cancer débute typiquement dans la peau, cependant aucune lésion cutanée n'est trouvée. Un *PET scan* ne permet pas d'identifier de nouvelles lésions. Dans une lettre, l'oncologue indique qu'il pense que la tumeur pulmonaire est une métastase d'un mélanome dont la localisation primitive (initiale) est inconnue. Ce patient est discuté en RCP et l'avis de l'oncologue est confirmé.

Pour cette question, la topographie choisie est `C80.9` (origine inconnue). Cette décision se fonde sur la morphologie de la tumeur (mélanome), sur le fait que les poumons sont des emplacements fréquents pour des métastases et suit les avis de l'oncologue et de la RCP. Les deux arguments faibles favorables $\left(\texttt{wp}_1^1, \texttt{wp}_1^2\right)$ et les deux arguments faibles défavorables $\left(\texttt{wc}_1^1, \texttt{wc}_1^2\right)$ associés à cette réponse sont :

$\texttt{wp}_1^1$ Un oncologue conclut que la localisation primitive est inconnue.

$\texttt{wp}_1^2$ Une réunion de concertation pluridisciplinaire conclut que la localisation primitive est inconnue.

$\texttt{wc}_1^1$ Un rapport d'imagerie indique une lésion tumorale dans le poumon gauche.

$\texttt{wc}_1^2$ Aucune lésion tumorale n'est trouvée sauf pour la lésion pulmonaire.

Le deuxième cas source $\texttt{srce}_2$ concerne une patiente pour laquelle un rapport d'imagerie indique une lésion tumorale dans le lobe supérieur gauche du poumon. Une biopsie de la tumeur permet de déterminer qu'il s'agit d'un adénocarcinome. Un test pour le marqueur TTF1 retourne positif. Un *PET scan* ne permet pas de trouver des lésions tumorales supplémentaires. En RCP, les cliniciens concluent pour une tumeur primitive du poumon.

Pour cette situation, la réponse choisie est `C34.1` (lobe supérieur du poumon). L'argumentation contient trois arguments faibles favorables à la solution choisie :

$\texttt{wp}_2^1$ Un adénocarcinome situé dans les poumons testant positif pour le marqueur TTF1 est un élément favorable à une tumeur primitive du poumon.

$\texttt{wp}_2^2$ Une réunion de concertation pluridisciplinaire conclut que la localisation primitive est le poumon.

$\texttt{wp}_2^3$ Aucune lésion tumorale n'est trouvée sauf pour la lésion pulmonaire.

A noter que $\texttt{wp}_2^3$ est le même argument que $\texttt{wc}_1^2$ avec un type d'argument différent.

Le troisième cas source $\texttt{srce}_3$ concerne une patiente. Suite à des douleurs abdominales persistantes, une imagerie est réalisée et le rapport décrit une lésion tumorale dans le côlon ascendant. Un *PET scan* permet de mettre en évidence des lésions tumorales additionnelles dans le lobe droit du poumon et dans le foie. Une biopsie permet de déterminer qu'il s'agit d'un adénocarcinome. Dans une lettre, l'oncologue conclut que la tumeur primitive se trouve dans le côlon et a formé des métastases dans le poumon et le foie. En effet, il est connu que les métastases d'un cancer du côlon peuvent se développer dans le foie et les poumons.

Pour cette question, la réponse choisie est `C18.2` (côlon ascendant). L'argumentation contient six arguments, quatre faibles favorables $\left(\texttt{wp}_3^1, \texttt{wp}_3^2, \texttt{wp}_3^3, \texttt{wp}_3^4\right)$ et deux faibles défavorables $\left(\texttt{wc}_3^1, \texttt{wc}_3^2\right)$:

$\texttt{wp}_3^1$ Un rapport d'imagerie indique une lésion tumorale dans le côlon ascendant.

$\texttt{wp}_3^2$ Un rapport de *PET scan* indique une lésion tumorale dans le côlon ascendant.

$\texttt{wp}_3^3$ Un oncologue conclut que la localisation primitive est dans le côlon ascendant.

$\texttt{wp}_3^4$ Un oncologue conclut que les tumeurs dans le poumon et dans le foie sont des métastases.

$\texttt{wc}_3^1$ Un rapport de *PET scan* indique une lésion tumorale dans le poumon gauche.

$\texttt{wc}_3^2$ Un rapport de *PET scan* indique une lésion tumorale dans le foie.

1.4.3 L'étape retrouver

Dans la méthode conçue pour ce projet, le but de l'étape retrouver est d'identifier le cas remémoré, c'est-à-dire le cas source dont l'argumentation est la plus pertinente pour résoudre le problème cible. Pour cela, un préordre $\preccurlyeq_{tgt}$ a été défini, utilisant trois critères $\mathcal{C}_{\texttt{strong}}$, $\mathcal{C}_{\texttt{weak}}$ et $\mathcal{C}_{\texttt{dist}}$ pour comparer et trier les cas sources par rapport au problème cible `cible`.

Arguments forts

Le critère $\mathcal{C}_{\texttt{strong}}$ utilise les arguments forts pour départager deux cas sources. L'idée de cet argument est de privilégier les cas sources pour lesquels il y a des arguments forts qui s'appliquent au problème cible. Étant donné un cas source avec un argument fort, si cet argument s'applique au problème cible et que le contexte général du problème cible est similaire à celui du cas source, alors la réponse à fournir est similaire à la réponse du cas source.

De par leur nature, les arguments forts sont rares. Ils sont utiles pour des situations plus faciles, correspondant à des règles et sont surtout utiles pour des encodeurs novices.

Soit $\Delta^{\mathtt{s}}_{\mathtt{tgt}}$ une fonction paramétrée par un problème `cible` qui prend deux cas sources et retourne la différence entre le nombre d'arguments forts favorables à ces cas qui s'appliquent à `cible`. $\Delta^{\mathtt{s}}_{\mathtt{tgt}}$ est défini comme

$$\Delta^{\mathtt{s}}_{\mathtt{tgt}}\left(\mathtt{srce}_i, \mathtt{srce}_j\right) = \mathcal{N}^{\mathtt{sp}}_{\mathtt{tgt}}\left(\mathtt{srce}_i\right) - \mathcal{N}^{\mathtt{sp}}_{\mathtt{tgt}}\left(\mathtt{srce}_j\right)$$

Ainsi une différence positive indique que $\mathtt{srce}_i$ est plus pertinent que $\mathtt{srce}_j$ pour résoudre `cible`. Une différence nulle indique qu'il n'est pas possible de trancher entre ces deux cas avec ce critère.

Dans l'exemple, il n'y a pas d'arguments forts favorables. Ainsi, $\mathtt{sp}(\mathtt{srce}_1) = \varnothing$, $\mathtt{sp}(\mathtt{srce}_2) = \varnothing$ et $\mathtt{sp}(\mathtt{srce}_3) = \varnothing$. Il n'est donc pas possible de trier les cas sources à l'aide de ce critère.

Arguments faibles

Le critère $\mathcal{C}_{\mathtt{weak}}$ utilise les arguments faibles, favorables et défavorables, pour départager deux cas sources, suivant la même idée que le critère $\mathcal{C}_{\mathtt{strong}}$. Cependant, il y a deux types d'arguments à considérer et il n'est donc pas possible de simplement compter les arguments applicables. En effet, il est préférable d'utiliser un cas source pour lequel deux arguments faibles favorables, ou encore pour lequel un argument faible favorable et un argument faible défavorable s'appliquent au problème cible plutôt qu'un cas source pour lequel deux arguments faibles défavorables s'appliquent. Pour prendre en compte cette particularité, un score a été défini pour estimer à quel point une argumentation peut être réutilisée. Ce score considère le nombre d'arguments faibles favorables et le nombre d'arguments faibles défavorables. Pour trier deux cas sources, ce score est utilisé, le cas source avec le plus grand score étant préféré.

Soit $\Delta^{\mathtt{w}}_{\mathtt{tgt}}$ une fonction paramétrée par un problème `cible` qui prend deux cas sources $\mathtt{srce}_i$ et $\mathtt{srce}_j$ et indique lequel de ces cas est préféré pour résoudre `cible`. Un résultat positif indique que $\mathtt{srce}_i$ est plus pertinent que $\mathtt{srce}_j$ pour résoudre `cible`. Une différence nulle indique qu'il n'est pas possible de trancher entre ces deux cas avec ce critère. $\Delta^{\mathtt{w}}_{\mathtt{tgt}}$ est défini comme

$$\begin{aligned}\Delta^{\mathtt{w}}_{\mathtt{tgt}}\left(\mathtt{srce}_i, \mathtt{srce}_j\right) = \quad & \lambda_p \cdot \left(\mathcal{N}^{\mathtt{wp}}_{\mathtt{tgt}}\left(\mathtt{srce}_i\right) - \mathcal{N}^{\mathtt{wp}}_{\mathtt{tgt}}\left(\mathtt{srce}_j\right)\right) \\ & - \lambda_c \cdot \left(\mathcal{N}^{\mathtt{wc}}_{\mathtt{tgt}}\left(\mathtt{srce}_i\right) - \mathcal{N}^{\mathtt{wc}}_{\mathtt{tgt}}\left(\mathtt{srce}_j\right)\right)\end{aligned}$$

où λ_p et λ_c sont deux coefficients positifs représentant l'importance des arguments favorables et des arguments défavorables. Dans l'approche actuelle, ces coefficients sont fixés à $\lambda_p = 3$ and $\lambda_c = 2$, afin de donner plus de poids aux arguments favorables. Lorsque plus de cas sources seront disponibles, il pourrait être intéressant de revoir les valeurs de ces coefficients.

Dans l'exemple, pour la comparaison entre $\mathtt{srce}_1$ et $\mathtt{srce}_2$, il faut d'abord identifier les arguments faibles applicables au problème cible. Ainsi $\mathcal{N}^{\mathtt{wp}}_{\mathtt{tgt}}(\mathtt{srce}_1) = 0$, $\mathcal{N}^{\mathtt{wc}}_{\mathtt{tgt}}(\mathtt{srce}_1) = 1$, $\mathcal{N}^{\mathtt{wp}}_{\mathtt{tgt}}(\mathtt{srce}_2) = 2$ et $\mathcal{N}^{\mathtt{wc}}_{\mathtt{tgt}}(\mathtt{srce}_2) = 0$. Ainsi $\Delta^{\mathtt{w}}_{\mathtt{tgt}}(\mathtt{srce}_1, \mathtt{srce}_2) = -8$. De façon similiaire, $\Delta^{\mathtt{w}}_{\mathtt{tgt}}(\mathtt{srce}_1, \mathtt{srce}_3) = -2$ et $\Delta^{\mathtt{w}}_{\mathtt{tgt}}(\mathtt{srce}_2, \mathtt{srce}_3) = 6$. Ainsi, $\mathtt{srce}_2$ est préféré à $\mathtt{srce}_3$, qui est préféré à $\mathtt{srce}_1$ pour la résolution du problème `cible`.

Dossiers patients

Le critère $\mathcal{C}_{\mathtt{dist}}$ n'utilise pas d'arguments et repose uniquement sur le dossier patient. Afin de départager deux cas sources, le dossier patient du problème cible est comparé avec celui des cas sources. Le cas source avec le dossier le plus proche de celui du problème cible est préféré. Comme les dossiers patients sont représentés par des graphes RDFS, une distance d'édition est utilisée pour déterminer le dossier le plus proche. La distance d'édition utilise l'approche définie dans la section 1.2.3.

Soit $\Delta^{\mathtt{d}}_{\mathtt{tgt}}$ une fonction paramétrée par un problème `cible` qui prend deux cas sources $\mathtt{srce}_i = (\mathtt{pb}_i, \mathtt{sol}(\mathtt{pb}_i))$ et $\mathtt{srce}_j = (\mathtt{pb}_j, \mathtt{sol}(\mathtt{pb}_j))$ et indique lequel de ces cas est préféré pour la résolution de `cible`. Un résultat positif ou nul indique que $\mathtt{srce}_i$ est préféré. Soit `dist` une fonction qui prend deux problèmes et retourne la distance d'édition entre les dossiers patients associés à ces problèmes. $\Delta^{\mathtt{d}}_{\mathtt{tgt}}$ est défini comme

$$\Delta^{\mathtt{d}}_{\mathtt{tgt}}\left(\mathtt{srce}_i, \mathtt{srce}_j\right) = \mathtt{dist}\left(\mathtt{pb}_j, \mathtt{cible}\right) - \mathtt{dist}\left(\mathtt{pb}_i, \mathtt{cible}\right)$$

Comparaison de cas

Pour comparer deux cas sources, le préordre $\preccurlyeq_{tgt}$ est utilisé. $\preccurlyeq_{tgt}$ utilise les trois critères $\mathcal{C}_{\mathtt{strong}}$, $\mathcal{C}_{\mathtt{weak}}$ et $\mathcal{C}_{\mathtt{dist}}$, dans l'ordre suivant, d'abord $\mathcal{C}_{\mathtt{strong}}$, puis $\mathcal{C}_{\mathtt{weak}}$ et finalement $\mathcal{C}_{\mathtt{dist}}$. Étant donné un problème cible `cible`, le cas source $\mathtt{srce}_i$ est préféré au cas source $\mathtt{srce}_j$ pour la résolution de `cible`, autrement dit $\mathtt{srce}_i \preccurlyeq_{tgt} \mathtt{srce}_j$, si

$$\begin{array}{ll} & \Delta^{\mathtt{s}}_{\mathtt{tgt}}(\mathtt{srce}_i, \mathtt{srce}_j) \ \mathtt{>}\ 0 \\ \mathtt{ou} & \left(\Delta^{\mathtt{s}}_{\mathtt{tgt}}\left(\mathtt{srce}_i, \mathtt{srce}_j\right) = 0\right. \\ & \quad \mathtt{et}\ \left(\Delta^{\mathtt{w}}_{\mathtt{tgt}}\left(\mathtt{srce}_i, \mathtt{srce}_j\right) > 0\right. \\ & \qquad \mathtt{ou}\ \left(\Delta^{\mathtt{w}}_{\mathtt{tgt}}\left(\mathtt{srce}_i, \mathtt{srce}_j\right) = 0\right. \\ & \qquad\quad \mathtt{et}\ \ \left.\left.\left.\Delta^{\mathtt{d}}_{\mathtt{tgt}}\left(\mathtt{srce}_i, \mathtt{srce}_j\right) \geqslant 0\right)\right)\right) \end{array}$$

Pour l'exemple, l'application des trois critères permet de trier les cas sources comme suit

$$\mathtt{srce}_2 \preccurlyeq_{tgt} \mathtt{srce}_1 \preccurlyeq_{tgt} \mathtt{srce}_3.$$

Donc $\mathtt{srce}_2$ est le cas remémoré pour le problème `cible`.

1.4.4 L'étape réutiliser

Une fois le cas remémoré identifié, celui-ci est utilisé pour fournir une solution pour le problème cible. Dans un premier temps, l'approche conçue dans ce projet utilise la même réponse et tous les arguments du cas remémoré qui s'appliquent au problème cible.

Pour l'exemple, la réponse fournie est `C34.1` (lobe supérieur du poumon), la réponse du cas $\mathtt{srce}_2$. L'argumentation fournie contient deux arguments faibles favorables, $\mathtt{wp}^2_2$ et $\mathtt{wp}^3_2$, car ce sont les seuls arguments de la solution de $\mathtt{srce}_2$ qui s'appliquent à `cible`.

Dans un deuxième temps, il est envisageable d'adapter la solution du cas source pour déterminer la solution du problème cible. Pour des questions de topographie par exemple, il pourrait être intéressant de modifier le code topographique à la situation décrite dans le problème cible. Ainsi dans l'exemple, la réponse du cas source $\mathtt{srce}_2$ référence le lobe supérieur, alors que pour le problème cible, il est question du lobe inférieur du poumon. Une adaptation consisterait alors à fournir la réponse `C34.3` (lobe inférieur du poumon). Des adaptations sont également possibles au niveau des arguments, en ajoutant par exemple des arguments provenant d'autres cas sources qui soutiennent la réponse fournie pour le problème cible.

1.4.5 Les étapes réviser et retenir

Les étapes précédentes se concentrent sur la résolution du problème cible. L'étape réviser a pour but de valider la solution fournie pour le problème cible. Dans l'approche actuelle, cette validation est manuelle et est réalisée par les experts de codification. Par la suite, lorsque les solutions fournies par le système seront plus fiables, cette validation devrait être plus rapide, ce qui permettra davantage de diminuer le temps de travail des experts de codification.

Pour l'étape retenir, le nouveau cas (`cible'`, `sol(cible')`) est revu pour évaluer s'il est intéressant de l'ajouter dans la base de cas. Le but de cet ajout est d'augmenter les compétences du système.

Afin d'éviter d'éventuels problèmes de performances pour l'étape retrouver, il peut être intéressant de ne pas ajouter un cas dans la base de cas. Il peut aussi être pertinent de revoir régulièrement le contenu des différents conteneurs de connaissances, afin d'assurer la qualité de leur contenu [Smyth, 1998].

1.5 Évaluation

Afin d'estimer la capacité de l'approche conçue dans le cadre de ce projet, une première évaluation a été réalisée. Cette évaluation préliminaire porte uniquement sur la validité des solutions fournies par le système. D'autres aspects méritent également une évaluation et pourront faire l'objet de travaux futurs, comme la facilité de compréhension des explications ou le gain de temps pour les encodeurs et les experts.

1.5.1 Méthode

Pour valider les réponses fournies par le système, un jeu de données a été collecté et validé par les experts du RNC. Cette base contient des questions réelles posées par les encodeurs du RNC portant sur la topographie de la tumeur à encoder pendant les années 2015 et 2016. Le choix du sujet de la topographie est motivé par la forte prévalence de ce sujet dans les questions posées et le nombre acceptable de réponses possibles (voir section 1.1.1). Ainsi, 38 questions et leurs solutions ont été formalisées. Dans les réponses, 28 codes topographiques différents sont utilisés, dont 6 codes qui sont utilisés dans plus d'une solution. Un total de 71 arguments sont présents dans les solutions, dont 61 ont pu être formalisés par une requête `ASK`.

Vu la faible taille de cette base, une validation croisée est réalisée (*leave-one-out cross-validation*). À chaque itération, un cas de la base est fourni au système pour résolution en utilisant le restant des cas comme base de cas. Pour évaluer la validité des solutions, le nombre de bonnes réponses et de bonnes argumentations est compté.

1.5.2 Résultats

Lors de l'évaluation, 10 des 38 cas ont reçu une réponse correcte. Parmi les 28 cas restants, pour 2 cas la bonne réponse est présente parmi les cinq cas sources les plus proches. Pour la partie concernant l'argumentation, 4 solutions contiennent les arguments attendus et 18 solutions ne contiennent aucun argument.

Les tableaux 1.2 et 1.1 fournissent le détail de des résultats. Dans le tableau 1.2, la première colonne (Id) identifie le problème. La deuxième colonne (Att.) indique la réponse attendue. La troisième colonne (Four.) indique la réponse fournie par le système et les colonnes suivantes indiquent les réponses des quatre autres cas sources les plus proches du problème cible. Les bonnes réponses sont soulignées. Dans le tableau 1.1, la première colonne (Id) identifie le problème. La deuxième colonne (Attendu) liste les arguments attendus et la troisième colonne (Fourni) liste les arguments fournis. Les arguments sont identifiés par un numéro et sont annotés par un astérisque s'ils sont formalisés par une requête `ASK`.

1.5.3 Discussion

Cette première évaluation a permis de valider le mode de fonctionnement général du système de résolution de questions décrit dans les sections précédentes. Cette évaluation a également permis d'identifier des pistes d'améliorations. A première vue, le nombre de réponses correctes paraît assez faible, cependant il est largement dû à la petite taille de la base d'évaluation et à l'approche actuelle de réutilisation. En effet, seuls 6 codes topographiques sont présents dans au moins deux cas sources. Pour les 22 cas sources pour lesquels la réponse n'apparaît dans aucun autre cas, en retirant ce cas de la base, le système n'est plus en capacité de répondre correctement. En augmentant le nombre de cas et en améliorant la méthode de réutilisation pour permettre l'accès à de nouveaux codes topographiques,

Id	Att.	Four.	Autres réponses			
1	C77.8	C48.2	C14.0	C67.9	C54.1	C34.0
2	C80.9	C80.9	C34.0	C56.9	C77.8	C34.1
3	C34.0	C34.0	C71.8	C48.2	C08.9	C77.8
4	C56.9	C48.2	C48.2	C57.9	C48.2	C80.9
5	C34.1	C34.0	C80.9	C14.0	C48.2	C71.9
6	C08.9	C71.8	C05.0	C48.2	C77.8	C34.0
7	C44.6	C50.5	C71.8	C80.9	C08.9	C34.0
8	C51.9	C60.9	C00.0	C67.9	C44.9	C44.0
9	C14.0	C48.2	C71.9	C56.9	C67.9	C21.1
10	C05.0	C08.9	C80.9	C71.8	C77.8	C56.9
11	C54.1	C11.0	C56.9	C80.9	C20.9	C60.9
12	C80.9	C56.9	C05.0	C11.0	C54.1	C14.0
13	C71.8	C08.9	C34.0	C77.8	C48.2	C05.0
14	C20.9	C08.9	C56.9	C11.0	C48.2	C54.1
15	C21.1	C20.9	C67.9	C44.9	C44.0	C71.9
16	C60.9	C00.0	C51.9	C44.9	C67.9	C44.0
17	C11.0	C08.9	C54.1	C56.9	C80.9	C20.9
18	C44.9	C00.0	C60.9	C21.1	C51.9	C67.9
19	C37.9	C77.8	C57.9	C71.8	C08.9	C48.2
20	C44.0	C67.9	C60.9	C21.1	C00.0	C51.9
21	C69.6	C44.9	C67.9	C00.0	C21.1	C60.9
22	C41.2	C56.9	C44.6	C48.2	C50.5	C80.9
23	C48.2	C48.2	C57.9	C56.9	C48.2	C41.2
24	C38.0	C80.9	C80.9	C71.8	C54.1	C11.0
25	C80.9	C80.9	C80.9	C48.2	C71.8	C50.5
26	C00.0	C60.9	C51.9	C44.9	C67.9	C44.0
27	C50.5	C71.8	C77.8	C80.9	C08.9	C05.0
28	C56.9	C57.9	C11.0	C54.1	C80.9	C14.0
29	C71.9	C67.9	C14.0	C21.1	C48.2	C44.0
30	C67.9	C67.9	C21.1	C71.9	C44.0	C44.9
31	C48.2	C48.2	C48.2	C14.0	C71.9	C21.1
32	C48.2	C48.2	C48.2	C48.2	C71.8	C57.9
33	C48.2	C48.2	C48.2	C71.8	C34.0	C08.9
34	C77.8	C56.9	C37.9	C71.8	C05.0	C08.9
35	C34.0	C34.0	C80.9	C14.0	C48.2	C71.9
36	C80.9	C38.0	C80.9	C11.0	C54.1	C48.2
37	C67.9	C67.9	C51.9	C44.0	C60.9	C00.0
38	C57.9	C56.9	C37.9	C71.8	C77.8	C48.2

Table 1.1: Détail des résultats pour les réponses fournies lors de l'évaluation.

Id	Attendu	Fourni
1	wp 1*,2* wc 3*	
2	wp 7*,8* wc 6*,9*	wp 8*
3	wp 58*,59*,60*	wp 10*
4	wp 13*,15*,62*,63* wc 61*	wp 109*
5	wp 18*,19* wc 20*	
6	wp 53*,54*	
7	wp 35*,36	
8	sp 48*	
9	wp 64*,65*	
10	wp 68*	wp 53*,54*
11		
12	sp 51	
13	sp 81*	
14	wp 69*,70*	wp 53*
15	sp 55*	wp 70*
16	sp 56*	
17	wp 83*,84* wc 85*,86*,87	wp 53*
18	sp 73*	
19	wp 75*,76	
20	wp 39*,40*,41*	
21	wp 77*	
22	wp 89*,90* wc 91*	wp 63*
23	wp 92*,93	wp 109*
24	wp 94*,95* wc 96*	
25	wp 8*,105	wp 117*
26	sp 79*	
27	sp 101*	
28	wp 103* wc 104*	wp 103*
29	wp 80	
30	wp 45*,108*	wp 45*,108*
31	wp 109*	wp 109*
32	wp 111*,112*	wp 109*
33	wp 109*,113*	wp 109*
34		wp 63*
35	wp 10*,115*	wp 60*
36	wp 116,117*,118*	
37	wp 45*,108*	wp 45*,108*
38	wp 13*,103*,119 wc 120	wp 13*

Table 1.2: Détail des résultats pour les argumentations fournies lors de l'évaluation.

le nombre de bonnes réponses devrait augmenter. En effet, en considérant seulement les 16 cas pour lesquels le code topographique attendu est présent dans la base de cas, 10 de ces 16 cas obtiennent une réponse correcte.

Pour des raisons similaires, les argumentations fournies ne contiennent pas beaucoup d'arguments. Cela est largement dû au fait que, dans la base d'évaluation, les arguments sont souvent présents dans un seul cas. Lors des itérations de l'évaluation, le cas n'est pas pris en compte dans la base de cas et donc il n'est plus possible pour le système de l'utiliser. Il pourrait être intéressant de permettre au système d'utiliser des arguments supplémentaires par rapport aux arguments utilisés dans le cas remémoré.

1.6 Conclusion

Dans ce résumé, la conception et la réalisation d'un outil d'aide pour le codage médical sont présentés. Dans un premier temps, les travaux réalisés sont destinés à faciliter la collecte des données pour les registres du cancer, cependant les résultats obtenus peuvent être appliqués à d'autres situations de codage. Cette collecte de données est souvent confrontée à des problèmes similaires. Pour comparer les données obtenues, il faut suivre les standards internationaux de codification. Comme ces standards couvrent de nombreuses situations, ils sont souvent complexes et difficile à appliquer. De plus, ils sont souvent complémentés par des recommandations dont le but est de combler les situations qui ne sont pas clairement présentés ou absents dans les standards. Tout cela représente un grand nombre de règles et consignes pour les encodeurs et pour les experts, qui manquent de solutions pour s'y retrouver.

Pour assurer la qualité de ses données, le RNC a mis en place une approche d'aide aux encodeurs sous la forme de formations continues mensuelles (ateliers) et de la possibilité de poser des questions de codification. Ces questions sont traitées par les experts de codification du registre, puis discutés lors des formations. Cette solution est coûteuse en temps et ne permet pas à elle seule de faciliter l'identification des règles ou consignes à appliquer.

Pour pallier à ce problème, ce projet a été initié avec les objectifs suivants :

- réduire le temps de traitements de questions pour les experts de codification,
- réduire le temps d'attente pour une réponse des encodeurs,
- faciliter l'identification de décisions de codification, de règles et de consignes.

Les travaux réalisés dans le cadre de ce projet ont porté sur la conception d'une méthode de résolution de question de codification utilisant du raisonnement à partir de cas et des arguments. Cette méthode a également fait l'objet d'une évaluation préliminaire et d'une implémentation dans un portail web permettant aux encodeurs du RNC de poser des questions. Cet outil est actuellement en test en interne au RNC et fera l'objet d'une phase pilote dans un futur travail.

1.6.1 Codification médicale

Il y a de nombreux travaux en cours dans le domaine de la codification médicale. L'un des apports majeur concerne la définition d'un vocabulaire commun pour la description de données médicales. Malgré cela, comme les informations sont partiellement présentes sous forme de texte libre, ce qui complique l'exploitation automatique de ces informations. De nombreux travaux de recherche portent sur l'analyse de ces textes en utilisant des techniques de traitement automatique du langage naturel (*Natural Language Processing*) [Stanfill et al., 2010]. Pour rendre le contenu des dossiers patients exploitable, une autre approche consiste à les structurer dès la saisie par le personnel médical. Cette approche présente d'autres difficultés, dont notamment l'énorme quantité de codes et leurs nuances ou encore l'évolution et la maintenance de ces codes. Certains travaux visent à aider cette saisie, en filtrant par exemple les codes affichés du contexte médical courant [Noussa-Yao et al., 2015].

1.6.2 Perspectives

Le travail réalisé dans le cadre de ce projet a permis de fournir une première version d'un assistant pour la codification de données médicales. L'évaluation réalisée a permis de mettre en avant certaines limites du système actuel, portant notamment sur la méthode de réutilisation. Dans un futur travail, il pourrait être intéressant de revoir cette approche pour augmenter l'expertise du système conçu. Parmi les pistes d'améliorations se trouve la possibilité de permettre au système de modifier la réponse finale en l'adaptant au contexte du problème cible. Ainsi, si dans le cas remémoré la réponse est le code du lobe supérieur du poumon et que dans le problème cible il est question du lobe moyen du poumon, il serait intéressant pour le système de pouvoir adapter la solution du cas remémoré pour indiquer l'utilisation du lobe moyen dans la solution au problème cible.

Une autre piste de travail porte sur l'utilisation des arguments et notamment sur la formalisation des autres parties non-formalisés dans l'approche actuelle. Ces connaissances supplémentaires pourraient permettre de fournir plus d'arguments. À terme, le système pourrait aussi remplacer des arguments non-applicables par d'autres arguments applicables pour mieux défendre la réponse choisie.

Une autre piste d'amélioration concerne l'évolution des standards de codification. Ces standards sont régulièrement mis à jour pour suivre les nouvelles connaissances médicales et les changements dans les maladies prévalentes. Pour éviter de devoir revoir tous les cas sources suite à une mise à jour, il pourrait être intéressant de modifier le système pour automatiquement appliquer les changements nécessaires lors de l'étape réutiliser. Pour cela, il faut décrire de façon exploitable tous les changements nécessaires. Or, ces changements sont souvent décrits sous forme de textes et une étape de formalisation s'impose. Néanmoins, cette capacité d'adaptation reste une fonctionnalité importante pour le système, car les registres sont prévus pour exister sur de longues périodes, et donc un changement de standards est inévitable.

Ce travail a porté dans un premier temps sur le registre national du cancer du Luxembourg (RNC). Pour la seconde version de ce système, il est prévu de généraliser d'avantage le système pour l'étendre à d'autres situations de codification médicale.

Chapter 2

Introduction

The world is a complex entity. In order to understand it, descriptions and observations are necessary. However, this is a very challenging undertaking. To begin with, it is necessary to know which features need to be observed and described. There are many aspects to consider to decide which features should be used, like

- current knowledge or model of the world,
- current resources and capabilities,
- goal and expected outcomes, *or*
- person tasked with choosing the features.

All of these aspects require serious consideration in order to obtain high quality, reusable and comparable descriptions. Fortunately the value of the resulting insights is often worth the hardship of this data collection task. This understanding of the world makes it possible for people to adapt to their surroundings. They can more easily predict problems and prepare for them, and sometimes even intervene, to make the world more hospitable for their way of living.

This general idea can also be applied to the health of the population. The domain which focuses on this topic is public health.

2.1 Public Health

Public health or population health is an area of activities focused on observing and improving the overall health of the general population. A healthier population is expected to be a smaller burden on the health care system. In fact, healthier people should require fewer treatments and drugs.

There are two main types of activities in this area, namely observation and intervention. These can be performed independently, but most often observation and intervention play a complementary role. Observation is used to assess priorities and identify measures for intervention. After these measures are implemented, observation is also used to evaluate the impact of these measures.

Observation entails activities related to measuring and reporting the health state of the population. It can cover the global population or focus on specific subpopulations. These activities are carried out by public and by private parties. Examples of public institutions include

- ministries,
- social security agencies,
- work safety and inspection agencies, *and*
- non-governmental organizations (NGO).

Especially in poorer countries, associations and foundations are also actively contributing to public health to compensate for missing public initiatives. There are different types of studies done for observation, like cross-sectional studies or longitudinal studies. Cohorts and registries are examples of long-term studies, usually meant to follow up on a given phenomenon on a global scale over a prolonged period of time. During these studies, data are collected on several occasions. They may serve for scientific publications or reports that could include recommendations of actions and changes to improve health for the target population.

Intervention entails activities related to changing the habits of people or the overall context (e.g. laws or culture), both in order to improve the health state of the target population. These interventions often result from prior observations. Similarly, both public and private initiatives coexist. There are different ways to promote the health of the population, like prevention campaigns or guidelines. Prevention campaigns may raise awareness of a given issue or may incite people to adopt new habits. In 2016, the Centers for Disease Control and Prevention (CDC) started the *National Prediabetes Awareness Campaign*[1] to raise awareness in the United States for people with prediabetes. The number of Americans with this condition are estimated to be around 84 million and most of them are unaware of their condition. In the campaign, a short survey is used to assess a person's risk of developing diabetes. A lot of resources have also been made available, informing on risk factors and possible lifestyle changes to prevent from developing type 2 diabetes.

Other types of interventions may provide additional support for people, providing them with additional or, more adapted resources in order to improve their living conditions. This may include facilitating access to drugs, medical experts, or social service to facilitate management of daily activities. Promoting research and industry through funding or other incentives is another possibility. For example, the European Commission offers prices for public and private organizations in the annual EU Health Award Ceremony. One of the prices rewards outstanding actions for the prevention of obesity in young people. In 2019, this price has been awarded to the city of Amsterdam for their *Amsterdam Healthy Weight Programme*[2].

In the European Union, cancer is one of the main causes of death [EuroStat, 2019]. Thus, in those countries, there is an increasing number of actions to identify causes and find most adequate treatments.

2.2 Oncology

Cancer is a group of related diseases, where cells start to divide and multiply in an abnormal manner [Stephens et al., 2009].

Different events may lead to the diagnosis of a cancer. For some cancer types, a patient will see a doctor after first symptoms have appeared, such as lingering pain, fatigue or loss of appetite. These symptoms may lead to a series of other exams to determine the cause. For the most prevalent cancer types, i.e. breast cancer for women and prostate cancer for men, regular screenings are being done for people at high risk.

The most common exam types used in diagnosis are imaging (e.g. CT scan, PET scan, X-ray), biopsies and blood sample analyses. Imaging exams are used to locate solid tumors. For solid tumors, biopsies can be used to identify the morphology of the tumor, i.e. the type and the behavior of the tumoral cells. During a biopsy, a tissue sample is taken from the suspected or confirmed tumor. This analysis is important to know whether the tumor is benign or malign for deciding on the treatment.

If previous exams are not sufficient, additional tests can be done. For instance, supplementary genetic or biological exams might be performed to identify specific genes or markers. These can be important for the choice of treatment, as certain markers or genes may be linked to a greater efficiency, and thus a better outcome for the patient. For instance, for lung cancer some mutations in

[1] `https://www.cdc.gov/diabetes/campaigns/national-prediabetes-awareness-campaign.html`
[2] `https://www.amsterdam.nl/sociaaldomein/blijven-wij-gezond/amsterdam-healthy`

the Anaplastic lymphoma kinase (ALK) gene have shown to be very receptive to treatment [Franco et al., 2013]. It is also important to know whether the cancer has spread to other body parts. Benign tumors do not spread into or invade other body parts. However, malignant tumors do, which can significantly reduce the chance of survival for the patient. Solid tumors can spread to other body parts when cancer cells become detached from the tumor and move to another location. There they start to develop secondary tumor, also called metastases. The first location where the tumor developed is called the primary location. Cancer can spread locally to adjacent body parts, through the lymphatic system to adjacent lymph nodes, or using the blood stream, to distant body parts.

With initial cancer diagnosis and the additional information on the extend of the cancer, treatment options are discussed for the patient. This discussion is preferably done in the context of a multidisciplinary team meeting, which is a meeting where clinicians with different specializations discuss cancer cases. Not all of the cases presented warrant extensive discussion, as for the most frequent cases the treatment guidelines offer a good solution.

Treatment options may take on several forms. For more benign tumors, no treatment may be necessary, only a regular surveillance to check for a possible malignant evolution. For malignant tumors, typical treatment options include surgery, i.e. removal of the tumor, chemotherapy and radiotherapy which aim at destroying the tumor cells. These treatments may be prescribed together, e.g. a chemotherapy, followed by a surgery and a radiotherapy. More recently other forms of treatment have been introduced, notably immunotherapy [Rosenberg, 2014]. Some forms of immunotherapy rely on boosting or interacting with the immune system of the cancer patient to reinforce its effectiveness against the cancer cells. Other forms use genetic inhibitors to interfere with the function of tumor cells, preventing them from communicating or replicating.

In some cases, the tumor can be removed in its entirety, with little impact on the patient, who may then resume their life normally. Unfortunately, this is not always the case. Cancer remains one of the major causes of death in higher income countries [World Health Organization, 2019], which explains the importance of the fight against cancer and its burden on society.

To understand how cancer is burdening the life of a patient and the health system in general, it is important to collect data on cancer, to gather information on incidence, diagnosis, treatment, deaths and risk factors. This is where cancer registries play a crucial role (detailed in section 2.3), as they aim to be an exhaustive and high quality database of cancer cases. With the landscape of cancer depicted by the collected data, public authorities and medical experts may then analyze the situation and define priorities and actions to fight cancer. These actions take on the form of public health policies, prevention campaigns, screening programs, treatment guidelines and validations or research incentives.

2.3 National Cancer Registry of Luxembourg

Registries are long-term studies meant to provide data on a given phenomenon. The aim of a registry is to enable the analysis of the trends of the given phenomenon, in order to assess its impact. Cancer registries are one example, focusing on the analysis of cancer and its burden on health care costs and living conditions.

In 2013, the Luxembourg Institute of Health (LIH) was mandated by the Ministry of Health to set up a nationwide population-based cancer registry in Luxembourg. The National Cancer Registry (NCR) is a systematic, continuous, exhaustive and non-redundant collection of data for each new case of cancer (excluding non-melanoma skin cancers). The NCR has been implemented according to international standards, recommendations and classifications. It is a large dataset with high level of quality, completeness and national coverage.

The aim of the NCR is to provide an objective analysis of cancer evolution in Luxembourg (incidence, prevalence, survival after a cancer diagnosis). It enables health care professionals and public authorities to better assess the quality of health care given to cancer patients. Another goal is to evaluate prevention campaigns and national cancer screening programs (i.e. for breast and colorectal

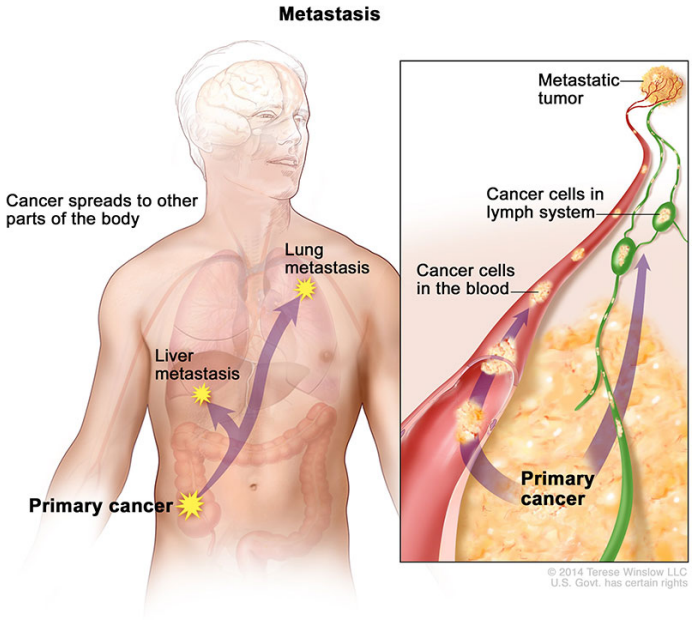


Figure 2.1: This image shows how a tumor can spread to other body parts using blood or lymphatic vessels. Source: `https://www.cancer.gov/types/metastatic-cancer`

cancers). On medium-term, the registry will serve as a tool to evaluate whether the National Cancer Plan is achieving its objectives.

All new cases of solid tumors (excluding non-melanoma skin cancers) since 2013 and of hematological malignancies since 2014 are recorded in the NCR. The NCR is a multi-source system and the main sources are the hospital-based cancer registries. The LIH has created a specific software for data entry, quality check and data export, called ONCOLIN, and has made it available to Luxembourg's four hospitals and to the National Radiotherapy Center (Centre François Baclesse). Other sources of data are the medico-administrative databases provided by the "Caisse Nationale de Santé" (national social security) and the "Contrôle Médical de la Sécurité Sociale" (health inspection). Data on the vital status of cancer patients is extracted from death certificates provided by the Ministry of Health. Pathological records from the "Laboratoire National de Santé" (national health laboratory) will be integrated into the NCR as soon as they are available in an electronic and standardized format.

Basic and advanced training courses on how to codify and introduce data from patient records into a hospital-based cancer registry are provided to data entry operators working at the hospitals (referred to as "Data Managers Cancer") by the team of the NCR. In addition, one-day workshops are organized ten times a year.

One characteristic that distinguishes the NCR from other population-based cancer registries, is the population it covers: the NCR includes not only people living in Luxembourg at the time of cancer diagnosis, but also people living abroad who have been diagnosed and/or treated in Luxembourg. Given the significant number of cross-border workers in Luxembourg and the European directive on cross-border health care, quality of care indicators and health care resource estimates that are based on the NCR data must take this specificity into account.

Specialized clinicians were involved right from the beginning of the NCR. Seven working groups of clinicians have been established. Of the activities carried out by these groups, clinical guidelines for prostate, lung, colorectal and breast cancers were prepared for Luxembourg, and then submitted for approval and publication to the "Conseil Scientifique dans le Domaine de la Santé". These working groups and the Scientific Committee of the NCR defined a set of quality of care indicators for different types of cancer (Breast, lung, prostate and colorectal cancers). The Scientific Committee is also responsible for the validation of all results before dissemination and publication.

NCR activities are conducted in close collaboration with hospitals, clinicians, the National Cancer Institute, foundations involved in cancerology, medical and scientific societies, and the Ministry of Health. Besides being a surveillance system, the NCR is recognized as an important member of the oncology landscape in Luxembourg. Representatives of the NCR participate in several national working groups within the framework of the National Cancer Plan, and in the scientific and technical committee of the National Colorectal Cancer Screening Program.

One of the purposes of the NCR is to provide an infrastructure dedicated to epidemiological and clinical research in oncology. One example of national collaboration is the future partnership with the Integrated Biobank of Luxembourg (IBBL) for the "Plan Cancer Collection" project (PKC project) within the framework of the National Cancer Plan. For this project, tumor specimens collected and stored at the IBBL will be annotated with data extracted from the hospital-based cancer registries and from the NCR, to create a national tumor bank.

By collecting standardized data with a high level of reliability, the NCR will be able to transfer Luxembourg cancer data to European and International organizations in order to compare results of Luxembourg with those of other European countries. The NCR is a member of the European Network of Cancer Registries (ENCR), the International Association of Cancer Registries (IACR) and the Group for cancer Registration and Epidemiology in Latin Language countries (GRELL).

The NCR has published its first national report on Non Small Cell Lung Cancer (NSCLC) quality of care indicators[3], in December 2018.

2.4 International Coding Standards

For public health, the collected data play a crucial role. In order to be able to compare the results of one study with results of another study, it is necessary to ensure that the collected data contain comparable information. This implies that the meaning of the collected features should be the same and that the process used to collect the data is similar (similar sources, surveys, exams, data cleaning and processing). However, given that these studies are often performed by researchers from different teams, institutions or even countries, it is necessary to have a global shared agreement of the previous aspects, i.e. international standards. The idea of standards is not specific to medical coding. In natural sciences, like physics or chemistry, all units have been defined in the International System of Units (SI), to facilitate sharing and comparing of measurements and results.

For medical coding, these international standards usually define the context in which the data are collected and used, in particular specifying the terms and vocabulary to use. The information is usually not kept in textual, but rather coded using alphanumerical sequences. For example, for the International Classification of Diseases for Oncology (ICD-O), which is a coding standard used for the registration and analysis of cancer cases, the topography of the tumor, i.e. the original body part in which a tumor started developing, is coded using a three digit sequence preceded by the letter `C` and a dot between the second and the third digit. `C34.1` is a valid topography code. There are a little over 300 topography codes, ranging from `C00.0` to `C80.9`. In general, one code may be linked to more than one medical concept. For the ICD-O topography codes, the code `C40.0` is used for all bones of the arms and shoulders. The number of codes and the grouping of concepts depend on the intended use of the data and is chosen so as to facilitate data analysis. This can lead to difficulties during the coding process, as the available source data might have been collected with a different purpose in mind. For example, the data collected in a patient record for clinical purposes (e.g. diagnosing and treating a patient) are different from the data collected for a cancer registry (observing cancer occurrences). The granularity might be different and the relevant information is not the same in both cases.

To overcome these problems, coding standards may provide some solutions and rules, however, they do not and cannot cover all possible situations. Thus to complete these standards, coding guides have been created. They provide additional guidelines and are not as strict as coding standards. They consist of expert knowledge, domain agreements and guidelines developed by everyday use. Their goal

[3] `https://www.rnc.lu/News/Article-RNC`

is to ease coding and increase the quality of the coded data, by facilitating the understanding and application of the sometimes very vast and intricate international coding standards. Some of these recommendations are also created and maintained by international organizations and working groups, like the recommendations for cancer registries of the ENCR[4].

As an illustrating example, let us consider the case of a particular male patient which should be coded for the NCR[5]. In 2013, he suffered from lasting pains in his side and a sudden loss of appetite. On January 12^{th}, 2014, a CT scan of his left kidney revealed nothing out of the ordinary. As the patient's condition continued to deteriorate, a second scan was made on February 15^{th}, 2014. This time, two suspicious neoplasms were found and the clinicians suspected cancer. Another CT scan made on March 10^{th}, 2014 showed signs of multiple renal adenopathy, which reinforced the cancer suspicion. On June 2^{nd}, 2014, a renal biopsy was carried out and the following histological findings pointed to a renal cell carcinoma.

To code this case, an operator needs to carefully read all the relevant parts of the patient record. For the NCR, there is a lot of data to collect. Some of it is mandatory and strictly defined by international coding standards. There are also data which have been selected by the various committees of the NCR. These data have been deemed useful for national indicators and measures. It is also possible to have data which are collected for a specific study, over a limited period of time. For example, a study on lung cancer might require more detailed information on the smoking habits of cancer patients than is normally collected for the NCR.

The mandatory data to collect concern the basic information about the cancer, like when (incidence date) and where (topography) it started, how it has been diagnosed and what type of cancer (morphology) it is.

The incidence date is the date of the event which allowed to confirm the cancer diagnosis. This date is usually not part of the patient record, as it serves no purpose from a treatment point of view. It needs to be determined by the operator of the NCR using the international definition of the incidence date provided by the ENCR[6].

The topography is the initial location where the cancer started out. This location is coded using the International Classification of Diseases for Oncology (ICD-O). In this coding standard, there are also rules which indicate how the correct topography code should be chosen based on the information present in the patient record.

The morphology describes the cell type of the cancer and how aggressive the tumor is expected to be. The cell type depends on the afflicted body part and the deterioration which caused the cells to develop a tumor. The aggressiveness indicates the potential of the cancer to spread to other body parts. A benign tumor for example does not spread to other body parts, unlike a malignant tumor. The morphology is also coded using ICD-O.

For the illustrating example, the operator determines that this type of cancer meets the inclusion criteria of the NCR and has to be coded. For this tumor, the incidence date is the February 15^{th}, 2014, the topography is `C64.9` (kidney) and the morphology is `M-8312/3` (renal cell carcinoma).

2.5 Coding difficulties

Coding is an essential step of the data collection process for cancer registries. In this step, coding standards play an essential role. However, the broadness and complexity of the standards can complicate the work of the operators. There are many types of cancer and each has its own specific aspects which are important to consider. In order to have a single standard which covers all of these, many different situations need to be accounted for and a common method needs to be found. Unfortunately,

[4] `https://www.encr.eu/recommendations-and-working-groups`

[5] In order to protect the privacy of cancer patients, all of the medical cases presented in this document were created for the purpose of explaining the work done in this project. Nevertheless, they realistically highlight the challenges faced by the NCR.

[6] `https://www.encr.eu/sites/default/files/pdf/incideng.pdf`

it is impossible for any set of rules to cover all the possible situations. Such a system would be too complicated. Thus the standards only cover important and simplified situations. In reality there are often nuances which will make the application of the standards difficult for operators. It is possible for example for data about one patient to be contradictory. One exam report might indicate a tumoral lesion in a given spot and another exam report might indicate that there is no lesion in that same spot. There are valid reasons for this situation to happen. In fact, as the cancer evolves over time, it is possible for a new tumoral lesion to develop in a different body part. It is also possible that one of the tests is less reliable or precise, thus leading to a wrong conclusion. In any case, this represents a challenge for operators, who need to determine which version should be kept and which should be discarded. This decision relies heavily on domain knowledge, like knowing which exam can be trusted for which information or which kind of evolution is possible for a cancer. While it is possible for an operator to acquire most of this knowledge, it still takes a lot time and practice, and it is an ongoing process. As new medical knowledge is discovered or coding standards change, an operator has to learn this new information.

Another difficulty faced by the operators of the NCR is the lack of possibility to specialize for a cancer type. In fact, by focusing on a smaller amount of cancer types, operators could limit how much knowledge they need to efficiently code cancer cases. This is not possible for the NCR, because the operators can only deal with the patients from one hospital and there are not enough cases and operators in each hospital to allow for specializations.

Another issue to deal with are coding inconsistencies. There are two types of inconsistencies, notably between different cancer registries and within one cancer registry. The first refers to the differences in interpretation of the coding standards that might occur. In fact, some parts of the coding standards may not always be very clear. These ambiguities might result in different understandings for different registries. Comparing the data collected differently by these registries may lead to the impression that there is a significant difference, like a higher incidence of lung cancer for the population covered by one of the cancer registries. The only way to deal with this situation is to identify the ambiguous rules and to provide additional guides or rules to prevent future misunderstandings. Recently, a difference has been noticed in how urothelial cancer is coded. To assess the extend of this issue, the ENCR contacted cancer registries in Europe and asked them to complete a survey on their registration practice for this type of cancer. The results of this study have been presented in 2018,[7] and resulted in the creation of a working group (*Urothelial Carcinoma Working Group*). This group has been charged with the creation of recommendations for cancer registries to clear up misunderstandings and provide additional explanations, in order to allow for more reliable comparison between the different registries and countries.

The second type of consistency is linked to coding decisions. As mentioned before, the coding standards do not cover all possible scenarios. For the situations which are not covered, the NCR has to provide guidelines detailing how they should be handled. This is necessary to guarantee the consistency of the data, meaning that the coding should be the same for identical situations. These new rules and guidelines need to be explained to operators. The increasing number of rules may further burden their work. Those situations are identified by the operators in their daily work and reported to the NCR in the form of coding questions. Each operator asks their questions individually and there is no platform to share them with the other operators or coding experts. To ensure that each operator is aware of these questions and their answers, the NCR presents and discusses them in the monthly one-day workshops, as these workshops are attended by all operators. It is also important that the answers provided to these questions are consistent, i.e. that two questions concerning similar patients receive a similar answer. As the questions are saved only in the exchanges with operators and in the minutes of the training, it can be difficult to determine if there has already been a similar question. At the moment, the NCR relies on the memory of the coding experts and operators, which may not always be a reliable solution.

[7] https://encr.eu/sites/default/files/2018-ENCR-Conference/Galceran_How%20can%20we%20improve%20and%20make%20more%20useful.pdf

2.6 Problem description and goals

This project was launched to tackle the coding difficulties faced by the NCR. The current approach requires a lot of time and resources to maintain the high quality of the data collected by the NCR and its operators.

More precisely, the problem addressed in this work concerns the work of a person tasked with extracting information from multiple data sources. Each data source can be seen as a set of facts or statements. For the application domain of this project, those facts or statements are the medical conclusions of clinicians and the observations found in the various exam reports. In the example from section 2.4, one observation is found in a CT scan indicating that two tumoral lesions (neoplasms) can be seen in the left kidney. Medical conclusions are done given the information available at the time and may for example concern the diagnosis of the patient's tumor.

The aim of this person is to interpret the data available and determine key information, as described in a guide or standard. This guide is accompanied by a set of rules and guidelines, that provide instructions for the most common situations on how to interpret the data coming from the different data sources. For the medical coding for a cancer registry, one such coding standard is the International Classification of Diseases for Oncology (ICD-O). This standard provides the codes used to describe the various topographies and morphologies of interest. In addition, some rules are provided to help an operator interpret the data available to them to decide on the code to use. For example, "Rule J", which describes how the order of words and synonyms should be handled to determine the adequate morphology code, states that

> *Compound morphology diagnoses: Change the order of word roots in a compound term if the term is not listed in ICD-O. Not all forms of compound words are listed in ICD-O-3. For example, "myxofibrosarcoma" is not in ICD-O-3 but "fibromyxosarcoma" is. The coder must check various permutations of the prefixes if the first one is not found.*

For the application, the aim of this project is to design and implement coding assisting, in order to reduce the time burden for the coding experts and operators of a cancer registry and to help ensure the coding consistency by facilitating the finding of similar situations. Initially, this work targets the NCR, however, it should be possible to apply this method to other cancer registries, and possibly even other disease registries. This coding assistant should serve as a tool for the NCR where operators can ask questions, get answers and find previously solved questions.

From a scientific point of view, the problem tackled by this project concerns the assistance of a person tasked with extracting information from multiple data sources, in a context where

- it is known which data are required,
- some of the necessary data might be missing, and
- some of the provided data might be contradictory.

The extracted information should follow the rules and guidelines provided in an evolving standard, which

- does not cover all possible situations,
- may change over time (e.g. new versions), and
- may grow over time to include more rules and guidelines.

The solution given by the designed method should also be explainable to users, to help them understand the solutions suggested by the method and serve as a learning tool for operators.

To achieve these goals, the current approach of the NCR, notably the coding support provided for operators, related work in medical coding and problem solving methods are reviewed. With this overview, a coding question solving method is designed, relying on case-based reasoning to answer coding questions for operators. The implemented method is also evaluated.

Chapter 3

Medical Coding Assistance

Medical coding is the process of reviewing textual medical documents and transcribing them using codes described in coding standards. This coding is necessary to follow up on the provided medical services, both in terms of quality and necessity. It allows for easier medical billing. The codes provided in the coding standards make it possible to focus on the relevant information, providing a level of abstraction for the content of the medical documents. In fact, even if two medical cases are presented slightly different, due to differences in the language used to describe them, the codes used are the same. This facilitates the analysis of these documents.

Given the huge amount of procedures, diagnoses and treatments, it can be very difficult for medical documents to be coded. Thus, support systems have been implemented, to help operators and other health professionals.

3.1 Current Work on Medical Coding

There is a lot of ongoing research in medical coding, notably on the creation and maintenance of coding standards, on coding support and on automated coding.

An important contribution of coding standards is the definition of common vocabulary and semantics. This is an essential element to obtain comparable data. To increase the quality and exhaustiveness of these standards, it can be very useful to include experts from various areas (e.g. different countries, hospitals) and domains (e.g. different specializations). This is usually done through the creation of working groups by international organizations, like the World Health Organization (WHO) or the ENCR.

When applying these standards, operators have to extract the relevant information and code it using the provided rules. However, the textual reports from the medical record may use different terms. It is possible that synonyms are used (e.g. influenza and flu), but more specific terms or more general terms (e.g. viral respiratory infection) could also be used. The information needed may also be split among several documents, thus needing some reasoning to reconstruct the information required by the coding standard. This partially explains the slow uptake of more automated coding systems, as both the difference in terms used and the lack of consistent structure constitutes a major challenge for systems.

To make the content of medical documents more accessible for machines, it would be interesting to structure them more precisely. However, this requires a huge amount of work for all parties involved. In fact, as medicine progresses and new discoveries are made, the items for each document need to be updated to follow the evolution of the field. The previously coded data might need to be updated and recoded using the new standards, which adds another burden for operators. It would also imply a major change for the daily activities of most health practitioners, with new tools and methods to support them in the use of these structured documents.

Another interesting avenue would be to enable machines to parse and use natural language text. This would reduce the changes in daily activities, as health professionals could continue to write free text reports.

3.1.1 Natural Language Processing

Natural Language Processing (NLP) is an area of computer science and linguistics that deals with parsing and exploiting data represented using natural language [Manning et al., 1999]. The complexity of NLP tasks comes form the richness of the languages used in our society. In fact, most languages are full of synonyms and expressions, allowing for very nuanced descriptions. In order to properly understand the content of a text, a system needs to know all of these synonyms and expressions, but it also needs a firm understanding of the context in which this text was written. Each domain can have its own specific ways and customs of describing and writing.

Despite this complexity, enabling machines to process natural languages provides some very interesting possibilities. Currently, there is a huge amount of information which exists only in textual form, with no specific structure. For example, the World Wide Web contains vast amounts of documents, with no coherent structure. There are some standards defining how to access these documents, create links between them and how to present them for a human user (e.g. hypertext transfer protocol [Fielding et al., 1997], cascading style sheets [Atkins et al., 2019]). Similarly, medical records for the most part are also only existing in the form of texts, with very little structure. For example, exam reports might have different sections (e.g. observations, conclusions), however, the content of these sections is a free text written by a health professional.

3.1.2 Automated Coding

NLP is used in many areas, medical coding is one of those. More recent research has focused on applying machine learning and other artificial intelligence techniques [Shi et al., 2017, Kavuluru et al., 2013a, Kavuluru et al., 2013b, Kavuluru et al., 2015, Pons et al., 2016, Stanfill et al., 2010] to parse and annotate medical documents automatically, minimizing human intervention as much as possible. The hope is to achieve at least human-like performance and benefit from the increased speed provided by computers.

In 2007, there was a contest at the BIONLP workshop for the annotation of ICD-9-CM[1] codes to radiology reports [Pestian et al., 2007]. For this task, a dataset of manually annotated, anonymized, English radiology reports was provided, with a learning set and a test set. The goal of the task was to add one or more ICD-9-CM codes to each report. Several designs were proposed [Aronson et al., 2007, Patrick et al., 2007, Crammer et al., 2007], some with very promising performance.

Despite recent progress, there does not exist any widespread solution for automatically coding documents.

3.1.3 Coding Support

Besides automatic annotations, there has also been research and development of tools to help operators and clinicians use the existing medical terminologies [Noussa-Yao et al., 2015]. When a user wants to assign a medical code for a given patient, most likely codes are presented, reducing the number of possibilities for the user. The challenge of this approach is the selection of the appropriate, most likely codes. To address this issue, probabilities could be used, as presented in [Lecornu et al., 2009].

Given the complexity of medical coding and in particular coding for a cancer registry, it is essential to provide support for operators.

3.2 Coding Assistant

To assist operators in their coding task, there are several possibilities. The NCR has chosen to allow operators to ask questions to its coding experts. When faced with a difficult situation to code, an operator can describe the situation and the difficulties faced. This description is forwarded to the

[1] International Classification of Diseases, 9^{th} Revision, Clinical Modification (ICD-9-CM) is an international standard for the coding of medical procedures and diagnosis, used for billing purposes.

coding experts of the NCR, which will answer the question. The most complicated questions are handled by the coding committee of the NCR.

This approach is also offered by other organizations, like the ENCR which allows cancer registries to ask questions through the website of the ENCR.[2] Similarly to the NCR approach, the question is asked using free text and forwarded to one of the appropriate experts of the ENCR.

Considering the coding difficulties faced by the NCR and the existing coding support system, it was decided to create a coding assistant to replace the current operator support system. Hence the goal of this project is to design and implement a method to answer coding questions asked by operators. The first application of this method is the NCR, however, it is planned to extend it to other registries, starting with other cancer registries and eventually also other medical coding instances.

With the new coding assistant, operators should be able to ask coding questions in a more structured manner. In the previous approach, operators could choose to use simple text, lists or raw exam reports to describe the problematic situations. While acceptable for coding experts, these different approaches make it difficult for a coding assistant to parse the descriptions.

When possible, the coding assistant should provide a tentative answer and an explanation, using the method designed in this project. The questions which could not be answered should be forwarded to coding experts. They should then use the interface provided by the coding assistant to answer these questions, relying on the structured description.

The coding assistant is expected to reduce the time burden of answering coding questions for coding experts. Given a question, the assistant attempts to answer it, freeing coding experts from handling some questions. The coding assistant is also expected to reduce the waiting time for operators, as an answer is immediately provided, albeit not always a perfect one. And finally, it should also increase the operators' understanding of the coding standards and coding guidelines.

3.2.1 Automated Coding for the NCR

For the NCR, the automated coding of the patient records is also an interesting solution to reduce the burden of coding. However, there are several challenges:

- no easy access to hospital patient records,
- no consistent patient record structure across hospitals,
- ongoing migration to electronic hospital health records, *and*
- hospital patient records are in German and/or in French.

Access to patient records for research purposes is always a delicate issue. The recent data protection laws in the European Union complicate this issue even further. The new law requires an explicit approval of the concerned patient to use their data, which would be very time consuming. Also, since the NCR is hosted by LIH, which is a separate institute and not part of any hospital, to gain access to the cancer patient records would require a formal collaboration with each hospital, in addition to patient approval. To circumvent individual approval, anonymized records could have been used. While there is ongoing work for anonymization of medical records [Szarvas et al., 2007], there is yet no easy, automated way to anonymize patient records.

Assuming that access to patient records is possible, parsing them presents another challenge. In fact, the records do not share the same structure across the four hospitals and the one radiotherapy center participating in the NCR. Thus, each hospital would require a separate parser with specific steps, in order to obtain a comparable information for the automated coding. Another possibility would be to create separate automated coding solutions for each institution, which would also increase the maintenance burden.

Another challenge is the ongoing migration of the current hospital records to electronic hospital records (EHR). In 2018, the four hospitals started to set up a new system to manage their hospital records, introducing electronic and structured records. Unfortunately, each hospital chose a different

[2] `https://encr.eu/ask-an-expert`

system, meaning that separate solutions for automated coding for the NCR might still be necessary. This process also started after the beginning of this project and it was not clear at the time when this migration would start and how long it would take before the data present in the new EHR would be ready for automated coding.

Another specificity of the NCR is the multilingual environment of Luxembourg. As most medical professionals are trained abroad, in France or Germany, and because there is no official ruling for the language to use for medical records, it is possible to encounter documents in French and in German. Thus, to apply NLP techniques, it is required to handle both languages, which would increase the difficulty of this task.

All of these issues would add a remarkable amount of work to this project, and largely exceed the available resources. Thus it was decided to start with a coding assistant. It can also be noted that it might be possible to reuse the coding assistant later to provide automated coding for the cancer registry, meaning that the work of this project will still remain valuable in the long term.

3.2.2 Implementation

To provide a solution that is sustainable and easy to distribute, it was decided to design the coding assistant as a web portal. The implemented application consists of a single page application (SPA) and a RESTful API with a backing triple store. The SPA has been developed using Angular[3] on top of the standard web application tools (HTML, Javascript, CSS). The backend has been developed using the Go language and the Gin framework[4] for the web server and the API. The triple store used is Apache Jena[5] with a Apache Fuseki[6] SPARQL endpoint to query the triple store. Authentication is managed following OpenID Connect[7] and OAuth 2.0[8] requirements and uses the IdentityServer 4 implementation.[9]

[3] `https://angular.io/`
[4] `https://github.com/gin-gonic/gin`
[5] `https://jena.apache.org/`
[6] `https://jena.apache.org/documentation/fuseki2/`
[7] `https://openid.net/connect/`
[8] `https://oauth.net/2/`
[9] `https://www.identityserver.io/`

Chapter 4

Case-Based Reasoning for Medical Coding

The target application of this research project is medical coding, in particular to facilitate the work for operators and coding experts. To achieve this goal, the current coding assistance provided by the NCR to their operators was analyzed. When faced with a difficult case, operators may ask questions using an online ticketing system on the web page of the NCR. These questions contain a partial description of the concerned patient, with optional anonymized exam results. Coding experts then read those and provide an answer, with an optional explanation. The applicability of this approach for other cancer registries was also reviewed and it was found that it could be applied. It was decided to create a coding assistant which would attempt to answer coding questions, thus easing the workload for coding experts and increasing access to help for operators.

While searching for suitable methods to solve the coding problems described in section 2.5, several key features or requirements were identified. A crucial requirement for medical coding is consistency. Similar cases need to be coded in a similar fashion in order to allow for an analysis of the coded data later on. For situations that are properly described in coding standards, there is little difficulty. However, for the remaining situations, it is important to both document and apply consistently any coding decision that was taken. Another important feature for the acceptability of this approach is the explainability of the question solving process. Coding decisions need to be explained in order to be understood by operators and by coding experts. This is important for later analysis, as how the data was coded can influence results. Explanations are also expected to increase user trust in the solution. The following sections show which methods and techniques were considered and chosen for this work, both for representing and solving the problems.

4.1 Explainable AI

As mentioned in the introduction, explanations are a key requirement of the method designed in this work, but also in many other applications [Gregor and Benbasat, 1999]. Explainable AI (XAI) is a recent and growing research field [Adadi and Berrada, 2018] in computer science aiming at creating artificial intelligence methods and algorithms that are understandable for a human user. The term was first used in [Van Lent et al., 2004] to describe their system. This trend is in opposition to the more classical "black-box" models like neural networks, who are inherently difficult to interpret. The interpretation consists of the process of analyzing (explaining) why the model chooses to provide a specific answer. There are numerous fields in which this feature is crucial [Vellido, 2019] from a user acceptance [Giboney et al., 2015] and trust point of view, and also from a legal perspective (e.g. accountability). There is indeed a growing number of laws which require information systems to be able to explain their actions, in particular if no human interaction is included in the decision process and the result may strongly affect the concerned person. This "right to explanation" is notably present

in the General Data Protection Regulation (GDPR) for the European Economic Area and has been present in French law since the 1970s[1].

There is a global assumption currently in the field of artificial intelligence that there is a trade-off between explainability and prediction prowess, i.e. that requiring an explainable solution would reduce performance. However, this assumption is not based on any objective study or data [Rudin, 2018]. Indeed, explainable models might even be safer, more reliable and accurate than current deep learning models and other "black-box" models.

There are two different approaches for XAI. One approach consists in setting up a second parallel system to explain the actual solving system, which will usually rely on traditional AI methods like neural networks. The major drawback of this solution is that explanations are produced independently of the actual solving process and might be different from the actual reasons used by the solving system.

In [Nugent et al., 2009], an explanation framework is added to the initial problem solving framework. For both, case-based reasoning is used. Case-based reasoning is not a "black-box" system, meaning it is possible to observe how a solution is computed. This is typically done by showing the retrieved case, i.e. the case which was used to solve the new problem. Nevertheless, while the retrieved case is typically the case that is most similar to the target problem, it might not be the best suited to explain the solution [McSherry, 2003a, McSherry, 2004]. In his work, McSherry proposed a method for selecting cases to explain the solution, focusing on finding cases which highlight the impact of the case features on the computed outcome.

In [Olsson et al., 2014], a method for explaining probabilistic models is proposed. In their example, they predict the energy performance of households. The actual energy performance prediction is provided by a probabilistic machine learning algorithm. In order to assist a user in the assessment of the provided solution, they show an estimation of the prediction error for the given problem. With this estimation, the user can more easily determine if the solution is reliable. Indeed, the higher the estimation error, the less reliable the predicted value is. To compute this estimated prediction error, case-based reasoning is used by aggregating the observed estimation error on similar cases.

Explanations are not limited to prediction systems, they are useful for other methods like recommender systems, which typically assist users in finding relevant content (e.g. items to buy, movies to watch, etc.). In [Gedikli et al., 2014], different explanation types are compared for a given recommender system.

Another approach consists in using a single solving system based on explainable reasoning methods. The major drawback of this solution is that they often require more knowledge and expert supervision to be built [Arp et al., 2014].

Currently most XAI systems provide a trace of their reasoning, rather than a model-based explanation as is usually the case in an academic context. Indeed, these systems focus on showing how they reached their conclusion, e.g. by using various visual aids, but for less expert users, those program traces are less likely to be useful. Even for domain experts, these traces may simply be too complex or too specific to provide any real insight into the model. There has been work done in the context of providing more high-level, model-based explanation, but they still require a lot of domain knowledge to be curated by domain experts and knowledge engineers, and this work is usually not easily transferable to other domains.

4.2 Knowledge Representation and Manipulation

In order to be machine-usable, knowledge needs to be described using a specific syntax, complemented with partial semantics. For that purpose, RDF is presented, as it has matured into a stable and widespread option, and there are several tools to manipulate it.

[1] *Loi no. 78-17 du 6. janvier 1978 relative à l'informatique, aux fichiers et aux libertés.* See [Bygrave, 2001] for more detail.

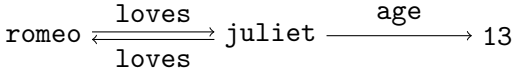
4.2.1 Resource Description Framework

Resource Description Framework (RDF) is a World Wide Web Consortium (W3C) specification used for data representation [Brickley and Guha, 2014]. It was initially intended as "a vendor-neutral and operating system-independent system of metadata" [World Wide Web Consortium, 1997]. The idea was to define a way of describing data and knowledge in a generic way, with multiple possible implementations, using XML, JSON-ND or Turtle syntax [Beckett and Berners-Lee, 2011] to represent descriptions. This effort resulted in a description framework relying on triples.

An RDF statement is a triple (`subj pred obj`) that can be understood as a sentence in which `subj` is the subject, `pred` (the predicate) is a verbal group and `obj` is an object. Thus (`romeo loves juliet`) is a triple stating that mister Montague has strong feelings for miss Capulet. An RDF base is a set of triples and is generally assimilated to a graph where nodes are subjects and objects, and edges are labeled by predicates. For example, the following RDF base

```
(romeo loves juliet)
(juliet loves romeo)
  (juliet age 13)
```

can be assimilated with the graph



and states that Romeo and Juliet love each other and that Juliet is 13 years old.

The Internationalized Resource Identifier is an internet protocol standard which extends the Universal Resource Identifier (URI) protocol. A IRI (and a URI) is a unique identifier for a resource, with a specific syntax for the identifier. They start with a protocol definition (e.g. `http`), followed by the separator `://` and finally a sequence of characters. The following are examples of valid IRIs:

- `https://en.wikipedia.org/wiki/Romeo` *and*
- `https://en.wikipedia.org/wiki/Juliet`

In addition, namespaces can be defined to serve as syntactic shortcut for manipulating IRIs. For example a namespace `wiki` can be defined as a shortcut for `https://en.wikipedia.org/wiki/` and the previous IRIs can be rewritten as `wiki:Romeo` and `wiki:Juliet`.

All elements described in RDF are called resources. Elements of an RDF formula are either IRIs or datatyped literals. In the previous example, `romeo`, `juliet`, `loves` and `age` are IRIs and `13` is a literal of datatype integer. Both subject and predicate of an RDF formula can only be IRIs, while objects can be either IRIs or literals. IRIs are used to identify resources described in an RDF base and can be referenced externally.

In some implementations of RDF, it is possible to define blank nodes, i.e. resources which cannot be referenced outside of the RDF base. They serve mainly as a syntactic shortcut, to prevent the explicit naming of local resources. For example, to describe that John owns a red sedan without explicitly referring to the exact instance of car, a blank node can be used to describe the car, rather than a IRI. Figure 4.1 shows a possible RDF graph using a blank node (represented with a circle). In several implementations for RDF, blank nodes are represented using brackets (`[ ]`). The following Turtle syntax describes the triples associated with the RDF base shown in figure 4.1:

```
john owns [
  a Sedan ;
  outsideColor red
]
```

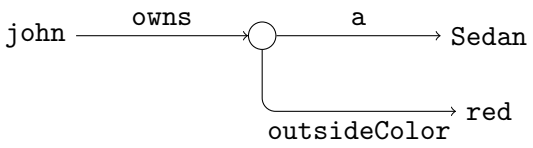


Figure 4.1: Example of an RDF base with a blank node, represented with a circle. This base describes that John owns a red sedan.

4.2.2 Resource Description Framework Schema

In RDF, there is no predefined meaning associated with any of the resources, except for their position within the triples. This limits the handling of the described data. To provide some standardized semantics, Resource Description Framework Schema (RDFS) was introduced.

In RDFS, some properties are associated with semantics, in particular to describe groups of resources and their relationship. The semantics are mostly used during inferences made when manipulating RDFS bases, i.e. mainly when querying the base. This allows a query engine to discover knowledge within the base that was not explicitly coded. This is a major difference compared to relational databases, where all information must be explicitly coded in order to be queryable later on.

A *class* is a special type of resource introduced in RDFS. Classes are used to group other resources that share similar properties or features. They can be understood as sets of elements, where the elements are resources. An instance is a resource which represents a specific element of this class. A class is itself a resource and belongs to the class of classes, i.e. `rdfs:Class`. For example, animals can be defined as a class for all instances of animals, like Tom the cat and Rex the dog. It is possible to define a hierarchy of classes to represent subsets of elements. For example, cats and dogs can both be defined as subsets of the class of animals. The subset class is a subclass and the superset class is called superclass or parent class. By convention in this document, resource names use only small letters to describe instances and capitalized names for classes.

To describe that a resource belongs to a class, triples are used with the property `rdf:type`, often abbreviated as `a`. For example, to say that Tom is a cat, the triple (`tom rdf:type Cat`) can be used, or more shortly (`tom a Cat`).

To describe the subclass relationship, triples are used with the `rdfs:subClassOf` property. For example, the triple (`Cat rdfs:subClassOf Animal`) indicates that cats are a subclass of animals. This property is transitive, meaning that if `a` is a subclass of `b` and `b` is a subclass of `c`, then it can be inferred that `a` is a subclass of `c`. Similarly, if a resource belongs to a class, it can be inferred that it belongs to each of its superclasses. For example, given the graph

$$\text{Tom} \xrightarrow{\text{a}} \text{Cat} \xrightarrow{\text{subc}} \text{Animal}$$

it can be inferred that

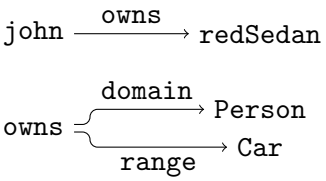
$$\text{tom} \xrightarrow{\text{a}} \text{Animal}$$

A *property* is a resource used to link other resources. All predicates in RDF triples are properties. All properties are of type `rdf:Property`, which is itself an instance of `rdf:Class`. Properties are used to describe features and relationships between resources. For example, in the triple

(`john owns redSedan`)

`owns` is a property.

To describe properties, two special properties `domain` and `range` were introduced. They provide typing information for the linked resources which can be used to infer resource typing. For a given property `p`, the `domain` property describes the type of the subjects used in triples with this property `p`. Similarly, the `range` property describes the type of the objects used in triples with this property `p`. For example, from the given RDF base



it can be inferred that

i.e. `john` is an instance of `Person` and `redSedan` is an instance of `Car`.

RDFS has some limitations, notably when it comes to additional constraints (e.g. number constraints).

There are more complex representation languages, which use a similar syntax, but rely on more complex logics, such as Web Ontology Language (OWL). This additional complexity however implies more complex reasoning engines with much higher computational requirements. In industry applications, those languages are often avoided if possible in favor of easier alternatives such as RDFS.

4.2.3 SPARQL Protocol and RDF Query Language

SPARQL Protocol and RDF Query Language [Harris and Seaborne, 2013] (SPARQL) is a manipulation and query language for RDF(S) bases. A query is usually composed of a specific action (insert, delete, select, ask, construct) and a graph pattern. The pattern represents a set of triples, with optional variables to replace any part of a triple. In SPARQL, variable names start with `?`, e.g. `?x` or `?tumor`. A variable can be matched against any element of the base. The pattern may also include some filters or other logical operations to limit possible matches found in the triple base.

In this research project, the most important type of SPARQL query used is `ASK`. This query tests the existence of a subgraph in a given graph, using variables. For example, the following query tests if someone (`?x`) in the queried RDF base loves a human (`?human`):

```
ASK {
  ?x loves ?human .
  ?human a Human
}
```

4.3 Semantic Web

In [Berners-Lee et al., 2001], the authors described their vision for the future of the World Wide Web. They argued that a lot of information is available through the internet. However, that information is often presented in very dissimilar ways, e.g. different vocabularies or technical implementations. There is also no clear indication of context. This makes exploitation of this information by machines an excruciating task. In order to tackle this issue, they imagined a standard approach for describing data and context, which would allow various agents to communicate and assist people in their daily activities. In their example, they envisioned an agent, which given the task to schedule an appointment with a family member at a nearby hospital, would look into their calendars, find a suitable time and given their location, would find a suitable hospital close by. In a review of the Semantic Web a few years later [Shadbolt et al., 2006], they noted that some progress had been made, and expected more to come in the future.

Linked Open Data [Bizer et al., 2011] (LOD) is an initiative to link the various knowledge bases which were created over the years. In order to facilitate the sharing of knowledge and data, Linked Data principles were defined. In a TED conference [Berners-Lee, 2009], Tim Berners-Lee defined these principles as follows:

- "All kinds of conceptual things, they have names now that start with HTTP."
- "If I take one of these HTTP names and I look it up...I will get back some data in a standard format which is kind of useful data that somebody might like to know about that thing, about that event."
- "When I get back that information it is not just got somebody's height and weight and when they were born, it has got relationships. And when it has relationships, whenever it expresses a relationship then the other thing that it is related to is given one of those names that starts with HTTP."

There are numerous ontologies representing medical knowledge, freely available like Medical Subject Headings (MeSh) or SNOMED Clinical Terms (SNOMED CT). They mainly define medical vocabularies, defining resources for many concepts (e.g. body parts, exam types, medical specialties, etc.) and some relationships between these concepts. Those relationships in particular are a very valuable resource. These ontologies, created before the LOD initiative, now also belong to the datasets of the LOD, allowing an automated system to access and use a considerable amount of medical knowledge in a standardized way. Storing this knowledge in a standard and clear location also facilitates the propagation of updates and additions to the knowledge of these ontologies.

4.4 Edit Distance

There are several ways to assess the difference or distance between two objects. One possibility is to use the cost of the changes needed to transform one object into the other. For example, the distance between the two strings `kitten` and `sitting` can be defined as `3`, as three changes are needed to transform `kitten` into `sitting`. These changes are

- replace the `k` with `s`: `sitten`;
- replace the `e` with `i`: `sittin`; *and*
- insert a `g` at the end: `sitting`.

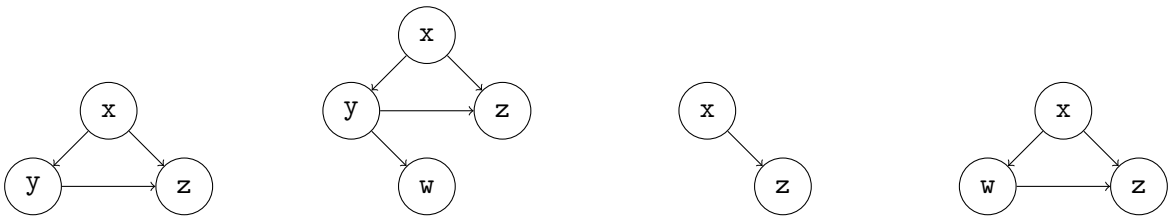
This distance is also called the Levenshtein distance [Levenshtein, 1966] and can be used on other types of sequences.

This idea of using changes to assess the distance between two objects can be generalized, in particular for graphs. The edit distance measures the difference between two objects `source` and `target`. This distance is not always symmetric, meaning that the distance from `source` to `target` might differ from the distance from `target` to `source`.

Graphs are mathematical objects used to represent objects and their relations. A graph is composed of nodes and edges. A node is an abstraction of an object. The relations between objects are represented using edges. An edge links two nodes of the graph together. An edge can be directed, meaning that a relation between two nodes `x` and `y` does not imply a relation between `y` and `x`. For instance, the relation "is related to" used to represent family ties is not directed. If Patrick is related to Silvia, then Silvia is also related to Patrick. However, the relation "is the child of" is directed. If Patrick is the child of Silvia, Silvia is not the child of Patrick. For a directed edge `e`: `x` $\rightarrow$ `y`, the edge `e` is called an incoming edge for the node `y`. If a graph contains directed edges, then it is called a directed graph. Otherwise, the graph is called an undirected graph. RDFS graphs, as seen in figure 4.1, are directed graphs. Both nodes and edges can be annotated with labels.

An edit operation is a single operation on `source`, meant to bring it closer to `target`. Edit operations commonly include insertion, deletion and substitution of a node or edge. An insertion adds an element (node or edge) from `target` into `source`. A deletion removes an element from `source`. A substitution replaces an element from `source` with an element of `target` of the same type (e.g. node for a node).

The cost of these operations can be constant or it can depend on the concerned elements. Typically, the insertion and deletion costs are constant, whereas the substitution cost depends on the concerned



(a) Initial graph. (b) Insert `w`. (c) Delete `y`. (d) Substitute `y` with `w`.

Figure 4.2: Examples of edit operations considering only nodes on a given graph.

elements. An important requirement, however, is that the cost of a substitution should be less than the cost of an insertion and a deletion. Otherwise, a substitution can be replaced by an insertion and a deletion to obtain a cheaper edit path, and thus making substitutions a redundant operation.

An edit path is a sequence of edit operations, meant to transform `source` into `target`. For a valid edit path, each node or edge can only be used in a single edit operation. If the edit operations include insertion and deletion, then it is always possible to find at least one edit path between two graphs. A trivial edit path consists in deleting all the source graph and inserting all the target graph. The cost of an edit path is defined as the sum of the costs of the edit operations that make up the edit path.

The edit distance from the graph `source` to the graph `target` is defined as the minimal cost of all valid edit paths from `source` to `target`.

Computing this distance is a complex problem, as the number of possible edit paths grows exponentially with the number of nodes in the compared graphs. Thus in practice heuristics are used to obtain reasonable performances, both for time and resources. In certain situations, it is possible to make some adjustments in how graphs are compared, in order to reduce the amount of edit paths to consider. For example, it is not required to take into account both nodes and edges when building an edit path. It is possible to consider only nodes or edges. In fact, if only nodes are considered, once an edit path is found, it is possible to deduce the missing edit operations on edges which will complete the transformation of `source` into `target`. Figure 4.2 shows examples of edit operations considering only nodes.

Another adjustment concerns trees, which are special cases of graphs. For trees, it is possible to reduce the number of valid edit operations, and thus decreasing the overall complexity. Given two trees `source` and `target` defined as

to compute the edit distance, the domain of possible edit paths is traversed. This is done by gradually building edit paths, until the path with the smallest cost is found. This search starts with an edit path with a single operation substituting $\mathtt{r}^s$ (root of `source`) with $\mathtt{r}^t$ (root of `target`). This edit path is then extended by adding edit operations which use only direct children of one of the roots of the compared trees, i.e. $\mathtt{c}_1^s$, $\mathtt{c}_2^s$, …, $\mathtt{c}_n^s$ and $\mathtt{c}_1^t$, $\mathtt{c}_2^t$, …, $\mathtt{c}_m^t$. For example, the edit path with a single substitution of $\mathtt{t}^s$ with $\mathtt{r}^t$ could be extended with a substitution of $\mathtt{c}_1^s$ with $\mathtt{c}_2^t$. When all of the direct children are used in an operation, this approach continues in a level-wise manner, by considering the nodes at the third level. This continues until the best edit path is found.

In order to further facilitate the computation of the edit distance, it is possible to limit substitutions to nodes of the same type. In the context of trees, the type of a node can be defined by the label of the edge of the parent-child relation. For example, in the graph shown in figure 4.1, the node `red` is considered to be of type `outsideColor`, and can only be substituted with another node of type `outsideColor`.

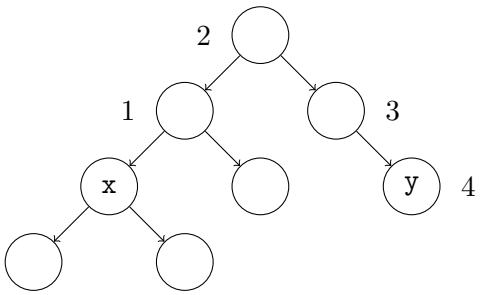
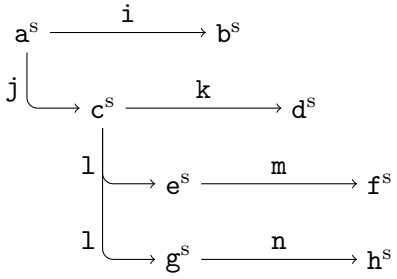
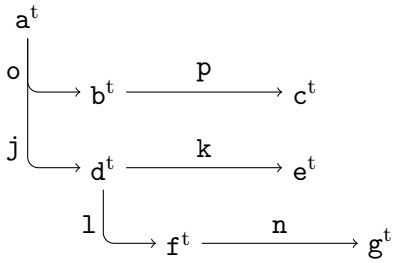


Figure 4.3: Small hierarchy with a representation of the nodes traversed to go from node x to node y. In this example, the hierarchical distance from x to y is 4 and the greatest distance is 5. Thus the substitution cost between x and y is $\frac{4}{5}$.



(a) Source graph.



(b) Target graph.

Figure 4.4: Example graphs used to illustrate an edit path.

Common edit operations are insert, substitute and delete. The insert operation adds a node from the target graph into the source graph. The delete operation removes a node from the source graph. Typically these operations have a fixed cost τ, which is a nonnegative parameter. The substitute operation matches a node from the source graph with a node from the target graph. There are different ways of computing the cost of this substitution. The cost usually depends on the label of the nodes. For labels that represent integers, a normalized difference can be used. For nodes that come from an ontology, e.g. RDFS resources, it is possible to use a normalized hierarchical distance. A hierarchical distance can be defined as the number of links traversed in the hierarchy to get from one node to the other. In order to have comparable substitution costs, this hierarchical distance can be normalized, using for example the longest possible path between two nodes of the hierarchy. Figure 4.3 shows an example of an hierarchy of nodes, as well as an example of a distance between two nodes x and y.

As an illustrating example of an edit path and the edit distance, let us consider the graphs shown in figure 4.4. To transform the source graph into the target graph, the following edit path can be used:

- substitute $\mathtt{a}^{\mathtt{s}}$ with $\mathtt{a}^{\mathtt{t}}$;
- delete $\mathtt{b}^{\mathtt{s}}$;
- insert $\mathtt{b}^{\mathtt{t}}$;
- insert $\mathtt{l}^{\mathtt{t}}$;
- substitute $\mathtt{c}^{\mathtt{s}}$ with $\mathtt{d}^{\mathtt{t}}$;
- substitute $\mathtt{d}^{\mathtt{s}}$ with $\mathtt{e}^{\mathtt{t}}$;
- substitute $\mathtt{g}^{\mathtt{s}}$ with $\mathtt{f}^{\mathtt{t}}$;
- substitute $\mathtt{h}^{\mathtt{s}}$ with $\mathtt{g}^{\mathtt{t}}$;
- delete $\mathtt{e}^{\mathtt{s}}$; *and*
- delete $\mathtt{f}^{\mathtt{s}}$.

Assuming that the cost of the edit operations used in this path is 1 for insertion and deletion and

(a) Source graph.

(b) Target graph.

Figure 4.5: Graphs used in the example for the edit distance between tree-like graphs. In the source graph, there is one "leaf" node that has two incoming edges, i.e. two parents.

`0` for substitution, the cost of the previous edit path is `5`. If there is no cheaper edit path, then the edit distance from the source graph to the target graph is also `5`.

4.5 Case-Based Reasoning

Case-based reasoning [Aamodt and Plaza, 1994, Richter and Weber, 2013, Riesbeck and Schank, 2013, Maximini et al., 2003] is a knowledge-based problem solving method, where in order to solve a problem, a previously solved problem which is similar to the new problem, is fetched. The method comes from the idea that often similar problems have similar solutions. Using domain knowledge, the solution of a previous problem should be usable to determine the solution of the new problem.

In a given application domain, a *case*[2] is the representation of a problem-solving episode. A case is usually represented by a pair (`pb`, `sol(pb)`) where `pb` is a problem related to the application domain and `sol(pb)` is a solution of `pb`. Given a new problem `tgt`, called the target problem, case-based reasoning aims at solving `tgt` by reusing a source case. A *source case* is an element of the case base. The *case base* is the set of previously solved problems with their solutions. A classical way to solve `tgt` consists in selecting a source case which is similar to `tgt` and to use it to solve `tgt`. The exact definition of *similar* depends on the application domain.

Case-based reasoning is a very general approach, which can be understood in different ways. The most classical approach uses a 4-R cycle, which stands for retrieve, reuse, revise and retain, and relies upon curated domain knowledge.

Case-based reasoning is a very versatile method and has often been combined with other techniques like rule-based reasoning [Chen and Wilkinson, 1998, Prentzas et al., 2008, Saraiva et al., 2015, Saraiva et al., 2016, Golding and Rosenbloom, 1991, Rossille et al., 2005, Surma and Vanhoof, 1995, Surma and Vanhoof, 1998], neural networks [Reategui et al., 1997] or preference-based reasoning [Hüllermeier and Cheng, 2013, Hüllermeier and Schlegel, 2011, Abdel-Aziz et al., 2013, Abdel-Aziz et al., 2014]. Given its intuitive nature and explainability [Cunningham et al., 2003], there have been several attempts at solving medical problems with case-based reasoning [Holt et al., 2005, Bichindaritz et al., 2015, Marling et al., 2014, Lieber et al., 2008].

4.5.1 The 4-R cycle

In [Aamodt and Plaza, 1994], the authors laid the foundation for the 4-R cycle found in many applications of case-based reasoning. Each iteration of this cycle represents one problem solving episode. The cycle, seen in figure 4.6, contains four main steps, i.e. the four "R"s: retrieve, reuse, revise and retain. The first two steps (retrieve and reuse) focus on the actual problem solving, while the other steps (revise and retain) focus more on the learning aspects. In these various steps, a lot of knowledge is involved. This knowledge is often classified into four different containers [Richter and Weber, 2013], as seen in figure 4.6, to distinguish between the different uses within the case-based reasoning systems. These four containers are called retrieval knowledge (RK), case base (CB), domain knowledge (DK) and adaptation knowledge (AK).

[2]The word "case" is used with different meanings in this document. To prevent possible confusion, when referring to a medical situation, "medical case" will be used instead.

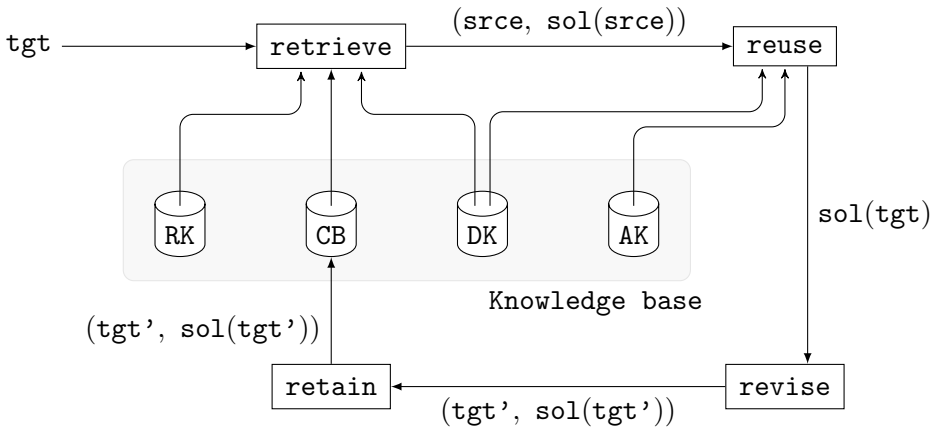


Figure 4.6: Case-based reasoning process using a 4-R cycle and knowledge containers.

Room features for hotel h_6

Amount	**Size (m^2)**	**Price ()**	**Price/m^2 (/m^2)**
10	15	45	3
10	18	72	4
3	25	150	6
2	50	300	6

Table 4.1: Example of a complete hotel description, as used for the computation of average price per night per square meter. Room prices represent the price per night.

When facing a new problem `tgt`, the first step is the retrieve step, where a similar source case is selected from the case base. In the next step (reuse), the problem and the solution of retrieved case are (re)used to solve the problem `tgt`. In the third step (revise), the problem `tgt` and its solution `sol(tgt)` are reviewed to verify for errors or other changes that could be interesting. In the final step retain, it is decided whether the new case (`tgt`, `sol(tgt)`) should be inserted in the case base (with possible changes).

As an illustrating example, let us consider the prediction of the number of stars for seaside hotels in a simplified setting. For this task, hotels will be described using two features. The first feature is the walking distance in tens of meters to the nearest beach. The second feature is the average ratio of the price per night and room size. A hotel can have an integer star rating ranging from 1 (worst) to 5 (best). In the case base, there are six hotels as shown in figure 4.7.

More formally, the problem domain will be composed of the set of hotels, a hotel being represented using a pair (`pm`, `d`), where `pm` is the average price per night per m^2 and `d` is the distance in tens of meters to the nearest beach. `pm` can be computed from the room descriptions by averaging the price per night per square meter of each room.

A solution to the problem will consist of a single integer, from the range 1 to 5 (both included). A case will be represented using the pair (`pb`, `sol(pb)`), where `pb` is a problem description, i.e. a hotel description, and `sol(pb)` is a hotel rating, i.e. an integer star rating. ((`6`, `2`), `3`) is an example of a case, for a hotel with a star rating of `3` located at 20 meters from the beach and with an average price per night per m^2 of `6` euros.

For the example, the target problem is the hotel h_{tgt} described by the pair (`3`, `5`).

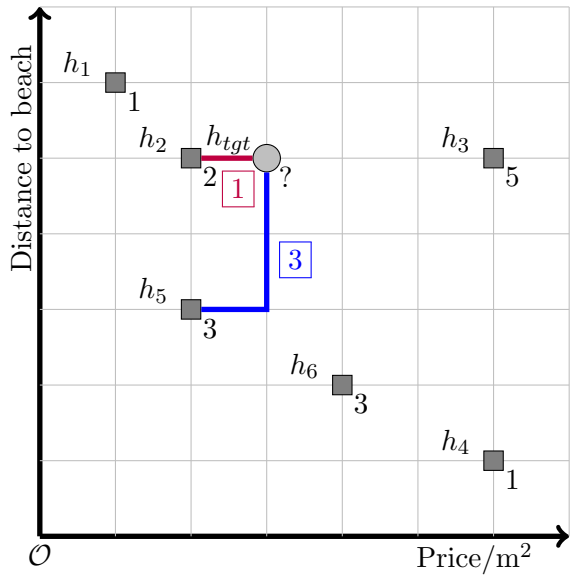


Figure 4.7: Case base for the hotel stars prediction task. Cases are represented using squares and the number of stars is shown next to the square. The target problem is marked with a circle. The colored lines visually show the Manhattan distance between the target hotel h_{tgt} and h_2 and h_5. The distances are shown in a box next to the lines representing the distance.

Retrieve

Each problem-solving episode starts with a description of the target problem `tgt`. This description is used in the retrieve step to find a similar source case, i.e. the retrieved case. There are many ways to identify this case. In some applications there may even be more than one retrieved case which will be used in the following steps of the 4-R cycle. To retrieve a source case, the concept of similarity is often introduced. If the problem `pb` of a case `srce` = (`pb`, `sol`(`pb`)) is similar to `tgt`, then it is assumed that the solution of `tgt` can be computed using `srce`. Similarity is often assessed using a measure. The exact definition and implementation of this measure depends strongly on the application domain and how problems are represented (e.g. vectors, RDFS graphs, etc.). It can be a simple weighted distance [Saraiva et al., 2016], a graph edit distance [Bunke and Messmer, 1993] or a more complex multistep method (e.g. MAC/FAC approach [Forbus et al., 1995]). Usually this similarity measure will rely on domain knowledge which is formalized by knowledge engineers and domain experts. There have also been attempts at automating the design of similarity measures using machine learning [Melacci et al., 2008].

For the illustrating example, a way to estimate the similarity between the cases in the case base and the target problem needs to be defined. For this example, after discussing and comparing several distances with domain experts, it was determined that the Manhattan distance is a valid option. The Manhattan distance is a sum of the absolute difference between each of the individual features of the entities. Given two hotels $\mathtt{h}_1 = (\mathtt{pm}_1, \mathtt{d}_1)$ and $\mathtt{h}_2 = (\mathtt{pm}_2, \mathtt{d}_2)$, this distance $\mathtt{d}_H$ can be defined as follows:

$$\mathtt{d}_H(h_1, h_2) = \mid \mathtt{pm}_1 - \mathtt{pm}_2 \mid + \mid \mathtt{d}_1 - \mathtt{d}_2 \mid$$

Figure 4.7 visually shows the distance between the target hotel and hotels h_2 and h_5.

To determine the retrieved case, the distance between the target problem and each source case in the case base was computed. The closest case will be used for the solving of the target problem, i.e. it will be the retrieved case. Table 4.2 shows the distance between each hotel and the target hotel. The hotel which is closest to the target hotel is h_2, with a distance of 1.

h	h_1	h_2	h_3	h_4	h_5	h_6
$\mathtt{d}_H(h_{tgt}, h)$	3	1	3	7	3	4

Table 4.2: Manhattan distance between the target problem and the various source cases.

Reuse

Once the retrieved case `srce` = (`pb`, `sol`(`pb`)) has been identified, the next step consists of using it to compute a proposition of the solution for the target problem `tgt`. This can be done in several ways, ranging from using a simple copy method to complex reasoning based on the solving process used for `pb`. As for the retrieve step, any complex reuse method will rely on domain knowledge obtained from domain experts.

For the illustrating example, a reuse by copy approach was used. This is the simplest reuse method, consisting of simply copying the solution of the retrieved case to solve the target problem. In this situation, the retrieved case is hotel h_2, which has a star rating of 2. Thus the star rating of 2 for the target hotel was provided, i.e. $\mathtt{sol}(h_{tgt}) = 2$.

There are more complex reuse methods, like relying on a majority vote using the `k` closest source cases, where `k` is a parameter which needs to be defined. If this approach were to be used with $\mathtt{k} = 5$, there would have been one source case with a star rating of 1, one with a rating of 2, one with a rating of 5 and two cases with a rating of 3. Thus the target hotel would get a rating of 3, i.e. $\mathtt{sol}(h_{tgt}) = 3$.

Revise

Once the target problem `tgt` has been solved, that is a tentative solution `sol`(`tgt`) has been provided, this new case (`tgt`, `sol`(`tgt`)) may need to be checked for errors or other necessary changes. This is the goal of the revise step. Using either or both manual and automatic methods, the correctness of the provided solution needs to be checked. Depending on the application domain, this validation may be needed to ensure the quality of the solutions. If there are known rules or constraints that the solution needs to follow, they can be implemented to ensure correct answers.

For the illustrating example, when reviewing this particular problem, after a visit of the target hotel, the domain experts conclude that the provided solution is incorrect. In this case, they can now correct it, replace the solution with the new rating of four stars, i.e. $\mathtt{sol}(h_{tgt}) = 4$. In the example, this error detection relies on domain experts.

For certain applications, it might also be useful to generalize the new case (h_{tgt}, $\mathtt{sol}(h_{tgt})$). In the example, instead of using only a specific price and distance from the beach, a problem description covering a region of the problem space, e.g. an interval of prices and/or of distances to the beach, in which all hotels have the same rating, could be used.

Retain

In the final retain step, it is decided whether the new, revised case (`tgt'`, `sol`(`tgt'`)), which is the corrected and validated case for the target problem `tgt`, should be included into the case based. By adding new cases to the case base, the case-based reasoning system is allowed to learn and potentially solve additional problems. New cases are expected to increase the solution coverage and quality of the reasoning system.

Depending on the number of cases and the content of the case base, it may not always be helpful to add a case. There have been studies to determine ways to assess if a case should be added into the case base [Smyth and Keane, 1995, Smyth, 1998]. It may also be useful or necessary to regularly evaluate and review the content of the case base to remove unhelpful cases, in order to improve performance, in particular for the retrieval of source cases.

For the illustrating example, it was decided to keep this new case, particularly since the provided solution has been validated by domain experts. This new case ((3, 5), 4) is added into the case base, which now has 7 cases.

4.5.2 Knowledge Containers

As described in the previous sections, several knowledge types are involved in case-based reasoning. This knowledge can be classified in four containers (seen in figure 4.6): domain knowledge (DK), retrieval knowledge (RK), case base (CB) and adaptation knowledge (AK).

The domain knowledge container consists of terms and definitions (i.e. vocabulary) which are relevant for the solving of problems in the given application domain. This type of knowledge is not specific to case-based reasoning.

The retrieval knowledge container consists of any information that is needed to compare cases and problems which each other. This knowledge is needed to assess similarity in the retrieve step of any case-based reasoning system. For example, if the similarity measure relies on a distance function parametrized by a tuple of weights, the weights associated with each feature are part of the retrieval knowledge.

The case base consists of all cases which can be used in a case-based reasoning system to solve new problems. These cases are usually build from past experiences, but may also be artificially constructed [Abidi and Manickam, 2002]. The latter often concerns domains where there are few solved past problems or where problem acquisition is very costly.

The adaptation knowledge container consists of information which can be used to adapt retrieved solutions when solving a new target problem. This information can take the form of rules. Adaptation methods can be used to reduce the size of the case base while maintaining good performances. Assuming that higher coverage of the problem space is key to a good performance, adaptation knowledge can extend the coverage of the source cases, thus requiring fewer to achieve the same coverage or increasing coverage with the same amount of cases.

4.5.3 Case Maintenance

Case-based reasoning is a learning system, i.e. it evolves over time to be able to solve more and more situations and/or provide better solutions. This is typically done by adding more cases or more knowledge. By adding cases into the case base, there is a risk to create performance issues. Adding new cases may not be very efficient, when they do not improve solution coverage. This can happen if the new case is very similar to an existing one and does not allow to solve any new problems that could not have been solved using an existing one. This could negatively impact the case-based reasoning system by reducing the performance of the retrieval step. To prevent this kind of errors, certain mechanisms can be put in place. A regular review of the knowledge containers, in particular the case base, can be performed, either by domain experts and/or by using an automated approach. There has been research into developing methods to identify issues and superfluous cases [Smyth and Keane, 1995, Smyth, 1998, Abdel-Aziz and Hüllermeier, 2015].

4.6 Other Problem Solving Methods

There are numerous methods for solving problems in a given application domain, case-based reasoning is just one of them. These methods have various strengths and limitations, and can sometimes also be combined to increase their efficiency. In the following paragraphs, some reasoning approaches and their possible synergies with case-based reasoning in the context of medical coding are presented.

For a running example, the coding of a patient for a cancer registry will be used. There are three reports for this patient, one imaging report, a surgery report and a histological surgery report. The first two reports indicate that there is a tumoral lesion in the pleura, spreading to the lung lobe. The last report indicates the cell type of the tumor and that there is a tumoral lesion in the lung lobe, spreading to the pleura. The report also contains a strong argumentation for the lung origin, based

on the tumor cell type. The information to be coded is the topography of the tumor, i.e. the location where the tumor originated.

4.6.1 Rule-Based Reasoning

Rule-based reasoning is a very intuitive approach, relying on rules to solve problems. These rules are usually obtained from domain experts with interviews, but can also be extracted from data directly or from documentation if the domain has existing models. Rules are simple constructs of the form of "if premise then consequence". If the premise is true in the context of the target problem, the instructions or conclusions provided in the right part of the rule can be applied. Depending on the application domain, rules can be more or less complicated and have more or fewer exceptions.

For the running example, the core issue lies in the contradictory information provided by the surgeon's report, the imaging report and the pathologist's report. According to the coding experts of the NCR, for the topography, the information from imaging reports and surgery reports should be preferred over histological reports, as the pathologist only receives a small portion of extracted tissue with a description of the origin of this tissue. The detected cancer type might not be sufficient to clearly specify a location, nor is it the purpose of a histological analysis to determine the point of origin, hence the rule

`if`	there is contradictory topography information between an imaging or surgery report and a histological report
`then`	the imaging or surgery report topography should be kept

Applying it to the previous coding problem, the topography would be coded as `C38.4` (pleura). However, in this example it was decided to follow the conclusion of the pathologist. So this problem falls into one of the exceptions of this rule, namely when the morphology of the cancer is not compatible with the provided topography. In this situation a different topography should be chosen. To account for this constraint, either rules or some other mechanisms to handle these exceptional situations could be added or the existing rule could be changed to check for compatibility between morphology and topography. For the second option, the rule could be changed to

`if`	there is contradictory topography information between an imaging or surgery report and a histological report
`and`	the topography provided by the imaging or surgery report is compatible with the provided morphology
`then`	the imaging or surgery report topography should be kept

In [Joseph et al., 2016], another project using rule-based reasoning is described. In their case, the goal was to identify the conditions of hospitalized patients by using their prescribed medications. Using rules crafted by experts, the described system used a very simple and elegant solution to identify most people.

Rules and cases can very naturally be seen as complementary, with cases representing specific episodes or instances (exceptions or yet unknown situations) and rules for global systematic situations. This complementarity can be seen very nicely in a project described in [Evans-Romaine and Marling, 2003]. The authors designed a system to assist medical students in their prescription of exercise training to patients. The implemented prototype uses both case-based reasoning and rule-based reasoning independently to solve problems. Then both solutions are presented side by side to show the difference between the guidelines, i.e. the rules, and exceptions or adaptations applicable to the current problem. The aim of this prototype was to teach these students how to adapt guidelines when needed.

There have been several attempts to combine both methods. In [Prentzas and Hatzilygeroudis, 2007], a review of various combinations is presented. In [Chen and Wilkinson, 1998], rule-based reasoning is used to reduce the number of candidate source cases, in order to decrease the memory use of their case-based reasoning system. Their idea results from the observation that for certain application domains, there is both subjective and objective knowledge that is used to solve problems. According to the authors, both types of knowledge are easier to represent using different models, rule-based reasoning for objective aspects and case-based reasoning for subjective ones. In [Prentzas et al., 2008], case-based reasoning is used in conjunction with symbolic rules. The main reasoning relies on rules, and case-based reasoning is used mostly to confirm the results of the rule-based system or to detect uncovered situations or known limitations (or errors) for rules. In [Saraiva et al., 2015, Saraiva et al., 2016], rule-based reasoning is used to enhance the performance of a case-based reasoning system. Their application domain is cancer diagnosis. In the retrieval step, the system relies on a weighted distance to compare source cases with their target problem. The proposed system uses rules defined with the help of oncologists to dynamically adapt the weights associated with various features, in order to integrate the difference of importance of these features for the different cancer types considered in their application.

4.6.2 Preference-based reasoning

There are many domains or situations in which a clear and definite solution cannot be provided. In those situations, either the best solution is not known or easily specifiable, or there simply is not an absolute "best" solution. The solution might depend on the actual user facing the problem, i.e. it's a matter of preference. For example for cooking, given a list of nutritional requirements, it is possible to find multiple recipes which satisfy those requirements, but it is difficult to compute whether a given recipe is "better" than another. This ranking will usually depend on the taste of the user. In a similar fashion, for traveling, given two perfectly acceptable destinations, a user might prefer going to warmer regions. Preference-based reasoning attempts to solve problems while taking into account user preferences. The exact nature of these preferences will vary from domain to domain (e.g. preferring warmer temperatures or sweeter meals).

In [Hüllermeier and Cheng, 2013, Hüllermeier and Schlegel, 2011, Abdel-Aziz et al., 2013, Abdel-Aziz et al., 2014], preference-based reasoning is presented as an alternative to classical case-based reasoning. The authors explain that in the classical approach of case-based reasoning, a single solution is retrieved to solve the target problem, which may not be realistic in some domains. There might be multiple optimal or acceptable solutions based on different source cases. However, in case-based reasoning, typically only the closest source case is kept and any additional potential source cases are ignored. The authors argue that these ignored cases could represent valuable information.

In the previous example, the challenge was to reconcile the different sources of information for the topography of the tumor, which provided conflicting answers. So it was necessary to choose which one, if any, would be preferred. This issue appears also for other types of information, e.g. the tumor size, where different types of exams or exam results at different moments often provide different results. Based on how the data will be used later and knowledge of the precision and context of the different exams, coding experts prefer to rely on certain exams rather than others. Often the option which indicates the worse diagnosis for the patient will be preferred. For the running example, imaging and surgery reports are usually preferred over histological findings for the origin of the tumor, as the pathologist relies only on cell type and the clinical description provided with the sample. The former cannot always point to a specific location, as the same morphology can appear in multiple body parts and the latter can be very vague and general.

4.6.3 Conversational Systems

Conversational systems are characterized by the different methods of interacting with their users. For conversational reasoning systems, the user is continuously involved in the reasoning process and helps

guide the reasoning process. This is typically meant to emulate the way a human expert would interact with a person, asking them questions and clarifications while the person explains their problems and needs. The problem solving system usually relies on a different, non-conversational technique. This technique can be any problem solving method, there is usually no need to adapt these (non-conversational) methods to work with a conversational system.

There have been many works [Aha et al., 2001, Branting et al., 2004, Folleso et al., 2014, Gómez-Gauchía et al., 2006, Gu and Aamodt, 2005, Gu, 2006, Gu and Aamodt, 2006, Mcsherry, 2001, McSherry, 2003b, McSherry, 2009, McSherry, 2011, McSherry, 2014] on the combination of conversational systems and case-based reasoning, i.e. conversational case-based reasoning.

In the previous example, a conversational approach would start similarly by asking the user which subjects to address and which cancer types are concerned. Then, the system would continue a "dialogue" with the user, and ask for information to specify the problem, requesting information based on the subjects and which information might be useful in this context. To determine the latter, it is possible to rely on the content of the case base, requesting information to either exclude potential source cases or find more similarities, or to rely on domain knowledge, if it is possible to define which information is needed to solve the problem. This domain knowledge could be extracted with the help of coding experts or, if sufficient cases are available, using data mining techniques.

4.6.4 Recommender systems

In retail, a lot of the services provided to clients are about guiding them to products they need and are more likely to buy. Recommender systems typically attempt to address this type of problems, where a user has a query or problem, which is more or less clear, and offers solutions to the user. This interaction can take many forms and can incorporate aspects of conversational systems (e.g. asking for more details for the problem description) or preference-based reasoning (e.g. choosing a solution based on user preferences).

Recommender systems have been used in the context of retail, assisting potential customers in finding products [Resnick and Varian, 1997, Linden et al., 2003], but also for help desk solutions [Wang et al., 2010] or movie suggestions [Gomez-Uribe and Hunt, 2015]. There have also been attempts at using case-based reasoning for recommender systems [Bridge et al., 2006, Jalali and Leake, 2012].

4.6.5 Belief Merging

In a perfect world, there would only be one truth and one model, and no information would ever contradict another without a valid reason. In reality, the situation is messier. Indeed, it is often the case that two sources of information will give you slightly different descriptions of the same facts. In some situations there is an explanation for these differences. It might be explained by a local or limited perception of the different sources. In other situations, the described phenomenon might simply have evolved or changed, resulting in different descriptions for the various states. For more subjective topics, there might simply be no way of verifying which statements are true and which are not. When merging those descriptions, the result will be contradictory, with no way to discard any of the statements.

Belief merging is a topic of logic which focuses on the handling of these types of situations [Gärdenfors, 2003, Katsuno and Mendelzon, 1991, Peppas, 2008]. In this area, models and methods to handle beliefs, i.e. information or statements, from various sources are studied, with the aim of creating one common set of beliefs (e.g. [Konieczny et al., 2004]). These sets of beliefs can be represented using sets of formulas. These techniques also have applications in other domains. There have been notably some attempts to combine case-based reasoning and belief merging [Cojan and Lieber, 2009, Cojan and Lieber, 2014].

In the context of the illustrating example, an application of belief merging could consist in considering every report as a separate source of beliefs, with the goal of merging them to obtain one location

for the described tumor. The merging would be guided by coding expertise. In the example, there are three sources, each providing a belief pointing to a given topography:

- imaging report: { topography = `C38.4` (pleura) };
- surgery report: { topography = `C38.4` (pleura) }; *and*
- histological surgery report: { topography = `C34.2` (middle lung lobe), morphology = `8825/1` (inflammatory myofibroblastic tumor) }.

Merging the first two sets of beliefs is trivial, but the third set contains a contradictory belief, i.e. about the topography. However, coding experts know that this type of cancer, i.e. this morphology, is not compatible with an origin in the pleura. There is no reason to reject this information about the morphology, as the source is a histological report, which is a trusted source for this type of belief. Thus to merge all beliefs, the pleura topography is dropped in favor of the lung topography.

4.6.6 Argumentation

Simply stating an answer or conclusion has rarely been sufficient to convince another person. It is necessary to discuss and address their current position in order to change someone's mind. This process of listening and discussing, also called argumentation, is not recent. Since ancient times philosophers have been practicing and perfecting this art. Formal argumentation is a research area which focuses on the formalization of argumentation methods [Caminada and Gabbay, 2009].

With the rise of decision support systems and the need for explanations, argumentation has started to make its appearance in artificial intelligence. One particular advantage of this method is that explanations are an inherent feature of the problem solving process. Already in the 90s, there were attempts at using argumentation in legal assistants. The HYPO [Ashley, 1991] and CATO [Aleven and Ashley, 1997] systems are two examples. These systems provide assistance for lawyers in the context of trials. There have also been attempts to use argumentation in case-based reasoning systems [Karacapilidis et al., 1997, Ontañón et al., 2015].

In the context of the illustrating example, the argumentation used to support the middle lung lobe topography is based on the strongly backed conclusion of the pathologist. In fact, in their histological report, they clearly state that they expected to see a tumor type which would indicate an origin in the pleura. However, the cell type they found is incompatible with this origin, and is in favor of a lung topography. In addition, it can be argued that when the imaging and surgery were performed, the tumor had already spread to both locations, making it difficult to identify the real origin.

Chapter 5

Case Acquisition and Representation

The application domain for this project is medical coding for cancer registries, in particular the NCR. To assist operators and coding experts in their task, a coding assistant was designed, relying on case-based reasoning to solve coding questions asked by operators. During the design of this system, previous questions were analyzed in order to understand the key features needed to answer a question. In this chapter, those features and how they are represented will be discussed, as well as how the necessary knowledge for the created assistant was collected.

5.1 Case Definition

In case-based reasoning, a case is a particular instance of a problem-solving episode. This definition is naturally domain dependent. For the application domain of this project, a case is a question asked by an operator and its answer. In the context of the NCR, operators are the medical staff employed by each hospital to code their medical cases of cancer. Each operator typically only has access to the medical record from their hospital. From this record, they extract all information relevant for the tumor to be coded. In order to identify which information is relevant for coding, questions addressed to the NCR over the last years (2013-2016) were reviewed and discussed with the coding experts of the NCR. From these discussions, it was determined that the relevant information concerns mostly exams and their main findings. To facilitate the question asking process for operators, the question form needs to remain as simple as possible, which implies that operators should not have to interpret exam reports, but only describe them as they present themselves. It was also concluded that for different cancer types (lung, breast, etc.), different additional information might be of importance, and that answers and arguments might also be slightly different.

As an illustrating example, let us consider the medical case of a patient John. In order to code his illness for the cancer registry, an operator will start by analyzing his hospital record. On January 17th, 2015, John was taken to the emergency room because he was suffering from abdominal pain. By chance, a suspicious opacity in his thorax was detected during his examination. In a follow-up CT scan on February 23rd, a tumoral lesion was detected, located just outside the middle lung lobe in the pleura. It measured 27 by 33 mm and infiltrated the lung. A PET scan on the 15th of March confirmed the tumoral finding and placed it around the lower lung lobe. A biopsy was performed on March 12th, however, the results were inconclusive (the cell type of the tumor could not be properly determined). On April 28th, John's situation was discussed in a multidisciplinary team meeting, and it was decided to operate his tumor. During his visit to the surgeon on the 1st of August to discuss the surgery, the surgeon indicated the preliminary diagnosis of a solitary fibrous tumor, i.e. a primary tumor originating in the pleura. On September 10th, 2015, John was hospitalized and the tumor was removed by the surgeon. In his report, the surgeon confirmed his diagnosis, noting that the tumor was found in the pleura and was infiltrating the middle right lung lobe. As part of the standard procedure, some of the removed tissue was sent to a laboratory for histological analysis. The laboratory report indicated that the tumor was removed entirely, however, the current diagnosis was questioned.

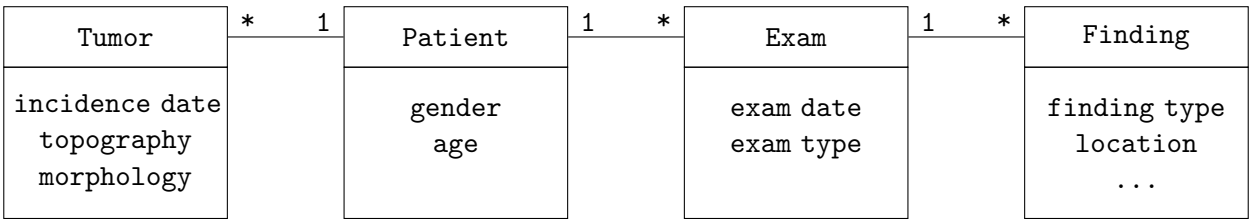


Figure 5.1: Relations between the entities present in this application, i.e. patients, exams and tumors. Each patient may be associated to any number of exams and/or tumors. Each exam may be associated to any number of findings. Each exam is associated to one patient, each tumor is associated to one patient and each finding is associated to one exam.

Indeed, the pathologist described clearly that given the cell type, this tumor actually originated in the lung and later spread to the pleura. Later in September, additional immunolabeling results came in, supporting the pathologist's opinion. Given the rare nature of this tumor, a second opinion by a university hospital was requested. On November 14th, 2015, the surgeon noted in a letter that John was recovering well and that no further oncological treatment was necessary, though a regular follow-up was recommended. On December 22nd, 2015, the pathologists from the university hospital analyzed the removed tumor tissue. Confirming the earlier suspicion, in their report, they indicate that they identified the tumor as a malignant solitary fibrous tumor, which is a rare mesenchymal tumor. In the actual reports, there are more details than in this summary, which can make the understanding of such a record quite difficult for an operator. In this case, there was a major contradiction for the topography of the tumor. It can be noted that for the surgery itself, given the close proximity of both locations, this distinction was not relevant, unlike for the cancer registry.

One of the pieces of information which needs to be coded for the cancer registry is the incidence date. As defined in the international standards, the incidence date is the date of the first event which enables the primary cancer diagnosis. There is a standard method to determine this date. To apply this method, an operator needs to collect the dates, types of exam, finding types and degrees of certainty (suspicion or confirmation) of every exam reporting a tumoral lesion. For the patient John, the operator needs the dates of the first visit to the emergency room, the different imaging exams, and the surgery. Similarly, for the topography of the tumor, the interesting information concerns every exam which refers to the tumor to code, the exam types and the exam findings.

The relevant information varies also with the overall cancer type. For example, for hematological cancer (blood cancer), detailed blood work is required in order to determine the topography and morphology of the tumor.

In order for the patient record description to be relatively close to the actual record, it was chosen to structure the information using the patient as a starting point. As shown in figure 5.1, the patient has some basic features like age and gender, and more complex features like the exams that were performed and tumor precedents. It was decided not to link exams explicitly to a tumor. In fact, for multiple tumors, linking exams to tumors would either require linking it to multiple tumors or copying the same report for every tumor. More importantly, it would force an operator to associate exams and tumors. However, if there is only one tumor, this information is of little use. If there are multiple tumors, then it is often the case that the operator does not exactly know which pieces of information should be used for which tumor, and thus forcing them to choose might introduce errors in the problem description, which will then need to be corrected by a coding expert. In addition, with a problem description which remains close to the patient record, it should also be easier to automatically extract some information from electronic patient records in the future.

With the coding experts of the NCR, a list of relevant exam types and findings was identified. To represent it, a combination of custom and existing ontologies was used. The encoding of body parts and morphology relied on the terms defined in SNOMED international version (SNMI). This choice was motivated by the desire to have a formal, structured description of exam findings which is as

close as possible to the original text in the exam report. This is mostly done in order to facilitate the description of the problem for operators and to prevent interpretation errors by operators. This also allows operators to ask questions even if they do not fully grasp the exam reports, in particular if they are not familiar with the terms used or if the information seems to be contradictory. Using SNMI also facilitates the mapping of the body parts and morphology terms to their respective ICD-O topography and morphology codes, as some partial mappings already exist.

For this project, a case consists of a problem description and a solution. The problem description contains the relevant information from the patient record for the coded tumor, a reference to the used coding standards, a general cancer type category and the question. The question is divided into separate subjects, each subject concerning different variables or information to code (e.g. topography, morphology, etc.). For each subject, an answer should be provided. By grouping various subjects in one problem description, the reuse of the patient record description is facilitated, i.e. an operator does not need to repeat the description for each subject.

In the solution, there is one answer and one argumentation (explanation) for each of the addressed subjects in the problem. The nature of the answer depends on the subject. For topography and morphology, the answer consists of an ICD-O code (e.g. `C34.1` for topography or `8140/3` for morphology). For cancer staging, the answer consists of the type of staging used (TNM or EOD) and the clinical and pathological staging. For the remaining subjects, the answer is a free text.

As for the argumentation, it is a set of arguments supporting or defeating the answer to the question. In the context of this project, an argument can be defined as a piece of reasoning. It explains how a coding expert decided which code should be used. As such, an argument can be split in three parts:

- the relevant information from the patient record,
- the medical knowledge used in the reasoning process, *and*
- the supported or defeated answers.

Arguments are not only used to explain the answer, but also by the coding assistant to answer questions. As an illustrating example, let us consider the argument stating that

> *A TTF1 positive adenocarcinoma found in the lungs favors the conclusion of a primary lung cancer.*

This argument can typically be used for questions about the topography of a suspected lung cancer. The relevant information from the patient record is the presence of the TTF1 marker and of an adenocarcinoma. To be able to use this argument, i.e. for this argument to apply, it is necessary to have information which indicates that there is a TTF1 positive adenocarcinoma. This result is provided by a histological analysis of a tissue sample, procured by a biopsy or a surgery.

The medical knowledge involved in this argument is based on research that has shown that many patients with a primary lung adenocarcinoma have been tested positive for the TTF1 marker. As the lung is a prime location for secondary tumors (metastases), this argument can help a clinician and an operator determine if this tumor should be considered as a primary location or a secondary location.

For this argument, the supported answers are the topography codes associated with a lung location, i.e. the codes `C34.0` to `34.9`. The exact code to use depends on other information present in the patient, usually an imaging that locates the tumor lesion.

For the illustrating example, the problem description can be structured as follows:

- **Subject**: topography
- **Cancer type**: lung cancer
- **Patient record**:
  - CT scan (February 23rd, 2015), with a finding of a tumoral lesion in the pleura;
  - PET scan (March 15th, 2015), with a finding of a tumoral lesion in the pleura;
  - Multidisciplinary team meeting (April 28th, 2015), with a finding of a suspicious tumoral lesion in the lower lung lobe;
  - Surgery report (September 10th, 2015), with a finding of a tumoral lesion in the pleura;
  - Histological report of the removed tissue (September 10th, 2015), with a finding of a tumoral lesion in the middle lung lobe and a finding of the morphology of the tumor as malignant solitary fibrous tumor.

The solution can be structured as follows:

- **Answer**: `C34.2` (middle lung lobe)
- **Argumentation**:
  - Weak pro: The pathologist indicates that the tumor originated in the middle lung lobe.
  - Weak pro: A malignant solitary fibrous tumor can develop in the lungs.
  - Weak con: A CT scan indicates a tumoral lesion in the pleura.
  - Weak con: A PET scan indicates a tumoral lesion in the pleura.
  - Weak con: A surgery report indicates a tumoral lesion in the pleura.

5.2 Case Representation

For any knowledge-based system, representation is an important issue. There are many different ways to represent information, with various advantages and disadvantages.

Attribute-value based representations are among the easiest forms of representation. However, one of their main limitations is the inability to represent links between entities. For this application, it is important to be able to link findings to their exams, and thus this kind of representation cannot be used, a more complex representation is needed. In particular for situations when the information provided by two findings is contradictory, knowing the exam which provided them is very valuable. Not all exams offer the same precision and reliability for certain types of insights and thus this can be used to decide which information should be preferred. For example, in order to determine tumor size, an MRI is more precise than an ultrasound.

Another form of representation used with case-based reasoning are RDFS graphs. They offer a very flexible solution and are very reliable in terms of available tools for management and storage. There are also many reasoning engines for RDFS, which can be leveraged for the design and implementation of case-based reasoning solutions. There are also a multitude of freely available, rich knowledge bases. The set of all of these bases compose the Linked Open Data, which is an initiative which was launched to unify and link the numerous datasets which were created after the rise of the Semantic Web.

For this application, RDFS has been chosen to represent the domain knowledge, including cases. When possible, concepts are taken from international and freely available ontologies. In particular, body parts and morphology concepts are taken from the SNOMED international version and its French counterpart [CISMEF, 2015]. Figure 5.2 shows the complete RDFS graph for the problem description of the illustrating example.

For arguments, there are several aspects to represent. An argument is an explanation for operators and coding experts. For that purpose, a textual description is used. This description is represented using a string. However, the coding assistant cannot rely on this text to answer coding questions, a formal representation is needed. There are three parts for an argument and each part requires a different representation. In a first stage, only the required relevant information from the patient record is represented for arguments. This is done by using a SPARQL ASK query. This choice makes it easier

to formalize which information is required. As described in section 4.2.3, an ASK query is used to identify specific patterns in an RDFS graph, which is exactly what is required to find the relevant information in the patient record. In order to formalize this part of the argument, it is only necessary to define which findings and exams are needed to apply the reasoning described in the argument. For example, the argument stating that

A CT scan indicates a tumoral lesion in the pleura.

relies on the fact that in the problem description there is a finding of a CT-scan which reports a tumoral lesion on the pleura. To check if the argument applies, it is necessary to verify that there is such a finding in the given problem description. This would result in the following SPARQL ASK query

```
ASK {
  ?exam examType ctScan ;
    finding [
      findingType tumoralLesion ;
      certainty certain ;
      location snmifre:T-29000
    ] .
}
```

which looks for an exam of type CT scan that found a tumoral lesion in the pleura, coded by the SNMI code `snmifre:T-29000`.

As mentioned in the previous paragraph, the two remaining parts of the arguments are not yet formally represented. Parts of the medical knowledge used can already be found in our various knowledge containers. They are represented using RDFS graphs. For instance, our knowledge containers include the complete list of body parts and a grouping into organs and body regions. This information can be used in the formal representation used to check for relevant information. In future work, it could be of interest to represent more knowledge and to exploit it.

The supported or defeated answers of an argument are also not yet represented. In the current version, this information can be partially found when an argument is used in a solution. If an argument is used as a pro in a solution, this means that the argument supports the answer of the question. However, this is not an exhaustive listing of the supported answers. Having the complete listing could be useful when building an argumentation, in order for the system to find new arguments which are not present in the retrieved case. This could be particularly useful to adapt an existing argumentation, if the initial arguments cannot be applied to the target problem.

5.3 Case Authoring

Unlike many other learning approaches like neural network variants, case-based reasoning systems can work without any prior cases. It is possible, though rarely done in practice, to start using a case-based reasoning system with an empty case base. In this situation, the system will most likely be unable to provide a solution or the provided solutions will be wrong. For both scenarios, a domain expert has to intervene to provide the missing solution. These problems and their solutions are then added to the case base, thus improving the competence of the case-based reasoning system. This approach is required if it is too costly or difficult to create a case base during the implementation of the system.

For this project, there are prior cases and thus it was decided to review and process some of the previously asked questions and their solution. During this initial case authoring process, with the invaluable help of the coding experts of the NCR, some domain knowledge needed for the knowledge bases of the coding assistant was also identified.

After the deployment of the designed system, the new problems and their solutions will be reviewed by the coding experts and will be integrated into the case base, to continue the learning process of the coding assistant. The reviews aim at validating the problem description and the solution. For

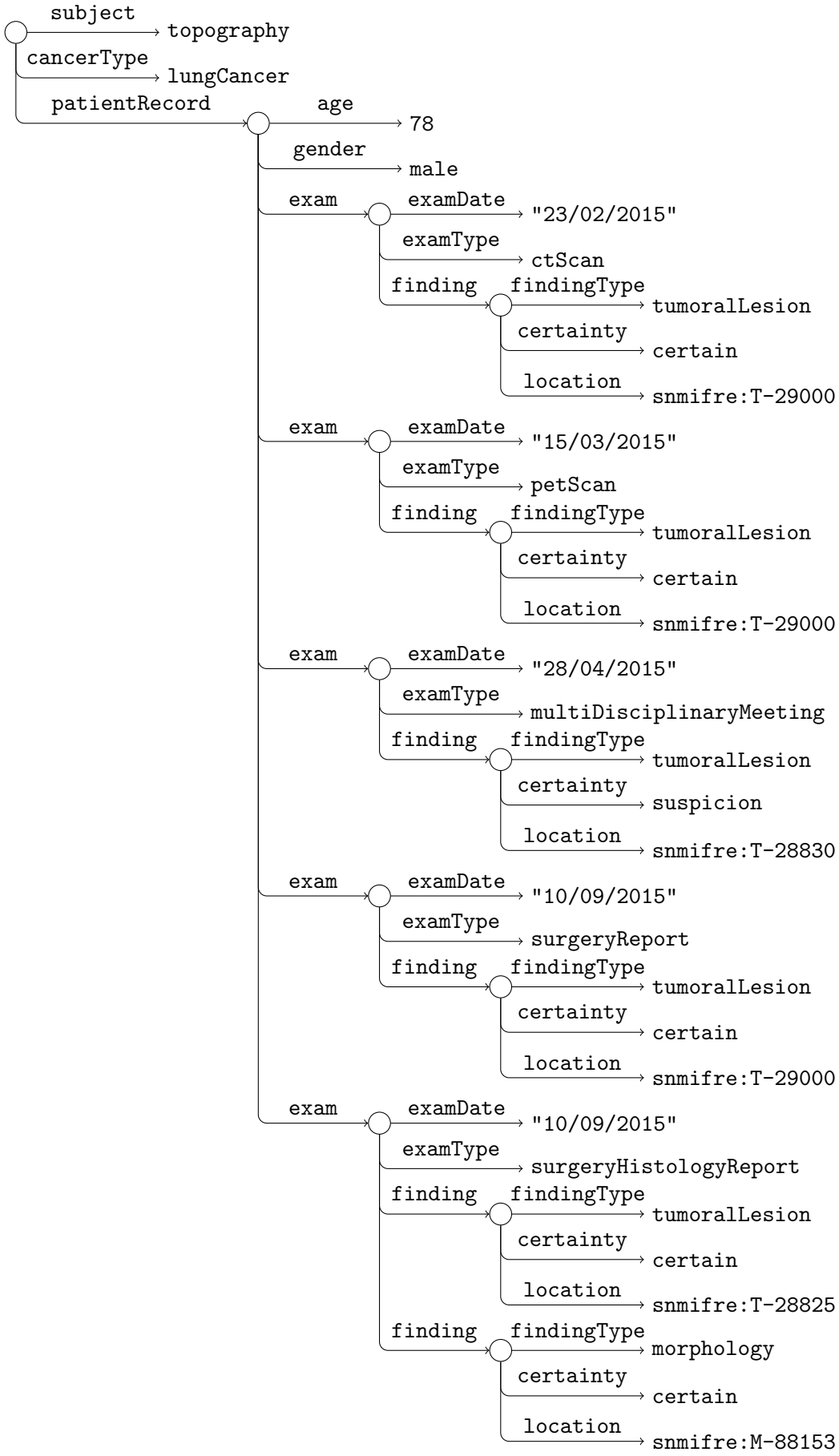


Figure 5.2: RDFS graph of the problem description of the illustrating example.

Subject	Year 2013	2014	2015	2016*	Total
Cancer staging (TNM, EOD)	37	71	43	16	167
Morphology	20	40	60	21	141
Topography	15	32	36	8	91
Multiple tumors/Inclusion	17	20	23	9	68
Incidence date	6	15	14	3	38
Other subjects	47	60	70	32	210
Total	142	238	246	89	715

Table 5.1: Evolution of the frequency of the top five recurring subjects in the coding questions from January 2013 to July 2016 (*: 2016 is only partially accounted for, which explains the drop in frequency).

instance, the problem description can be modified to remove useless information, the answer can be corrected or arguments can be added to complement the argumentation.

5.3.1 Initial Case Acquisition

To help jump start the system designed in this project and produce better results from the beginning, prior questions and their solutions were reviewed using the documents of the NCR. Since the start of the NCR in 2013, there have been monthly workshops for the operators of the registry. During these workshops, the coding experts of the registry share new information about the registry itself, the progress of the data collection and analysis, and most importantly, changes in coding practices. In particular, questions asked and solved by the experts are presented and discussed, to ensure that each operator will be familiar with these more difficult situations and knows how to code them. By implicating the operators, the NCR also ensures that any knowledge or expertise that the operators may have can be used when handling those difficult cases. Indeed, it is possible for the team of the NCR to have overlooked some previous similar situations or to have a different understanding of it. During these discussions, operators may bring up any concerns or remarks, which should increase the quality of the coding solutions. This sharing of questions and their solutions is also important for situations which are not covered in the coding standards or coding manuals provided by the NCR. The NCR can also use this opportunity to highlight common errors seen during data cleaning. For example, if the NCR notices that for a specific type of cancer the wrong topography is often used, they can discuss it with operators to understand where this misunderstanding comes from and explain why a different code is more appropriate.

For each workshop from January 2013 to June 2016, the workshop reports and any attachments regarding coding questions were reviewed. First, the subjects in those questions were analyzed. In table 5.1, the subjects encountered over the years and in total are reported. Cancer staging (TNM and EOD – Extend of disease), morphology and topography are the most frequent subjects. To limit the scope of the project, the coding assistant should only provide solutions for the three most frequent subjects.

The initial case acquisition focused only on topography questions. This choice is motivated by the importance of this information for the cancer registry and the relative ease of definition of the required information form the patient record. Indeed, the topography is one of the mandatory information for any tumor in a cancer registry. The topography is also a decisive factor for the selection of the information to be coded for the NCR.

The authoring of the cases was a multistep process for each question asked. It started with the review of the questions, including any attached exam reports and other remarks. From either the textual description given by the operator or the attached reports, the relevant information was

identified and transformed into a structured problem description in the form of an RDFS graph. This was done manually at first and later using the developed interface. Using the interface also served as a way to validate that operators can properly describe their questions in the implemented system. During this process, missing or incomplete features in the interface were corrected. Then the solution and arguments described in the workshop report were added. With the help of the coding experts of the NCR, the problem descriptions were reviewed, making sure that all relevant information was present and, if necessary, the argumentation was completed. At this stage, new arguments would consist only of the textual description of the argument. This description is meant to be shown to operators as an explanation. In the next step, new arguments were reviewed to provide a formal representation of the argument, when possible. These SPARQL ASK queries are added to the solution of the problem. In order to prevent the writing of overly complex queries, it was decided to provide a simple formalization for the arguments. Using this simplified approach can introduce some errors later, as the system might suggest arguments which are not applicable. However, at this stage it seemed more relevant to have simple arguments which can be reused easily. Also for some arguments, the amount of context and conditions which need to be asserted is immense, as a lot relies on "common sense" in the coding domain, which can be difficult to represent. In addition, when faced with a new problem in which there is an argument which should be not applicable, this argument can then be reviewed to restrict its applicability. This iterative process will allow us to more precisely identify which conditions should be added.

The codes used in medical coding have different precision levels. This is the case for topography codes and for most of the remaining information that is coded for the NCR. A specific code is always preferred, but unfortunately it cannot always be applied. Figure 5.3 shows the exact topography codes and the corresponding part of the colon. As an illustrating example, let us consider a medical case of colon cancer where there is very little information on the exact part of the colon in which the tumor originated. In this situation, the generic code `C18.9` (colon, not otherwise specified) needs to be used to code the topography of the tumor. The arguments which support this kind of decision (using a more general code) have proven difficult to formalize. The argument stating

> *If no precise location for this colon cancer can be determined, then the generic colon location should be used.*

is an example. With the chosen formalization of the arguments, the formal representation of this example needs to match patient records for which there is a colon cancer, but for which the exact location within the colon is unknown. For that, it is necessary to know which code is a generic code and which one is specific, and to know which location of the tumoral lesions described in the patient records matches a generic or specific code. For other cancer types, there might be additional ways of identifying the topography, which further complicates the formalization of these arguments. To support a generic code, it is necessary to show that none of the specific codes can be applied.

5.3.2 Reviewing and Revising New Cases

In the previous section, the initial case authoring was described. However, as the coding assistant is fed with new questions, the answering process may lead to the creation of new cases to be added to the case base. During the revise and retain steps of the designed application, the questions asked by operators and the solutions given by the system or by the coding experts are reviewed. Especially in the early phases of the system, each new case obtained is reviewed with the help of the coding experts and is prepared to be added to the case base. This process should however be quicker than the authoring of a case from the workshops. Indeed, the problem description will already be properly formatted. It is only necessary to ensure that all the relevant information is present. Some of the useless information can be removed, to create more general problem descriptions. For the solution, coding experts need to validate the provided solution, the argumentation, completing it if necessary and validate the used source case. For the argumentation, if there are new arguments or changes to be made to existing arguments, these need to be made as well in the formal counterpart, i.e. the SPARQL ASK queries. This formalization will happen only for subjects which should be answered

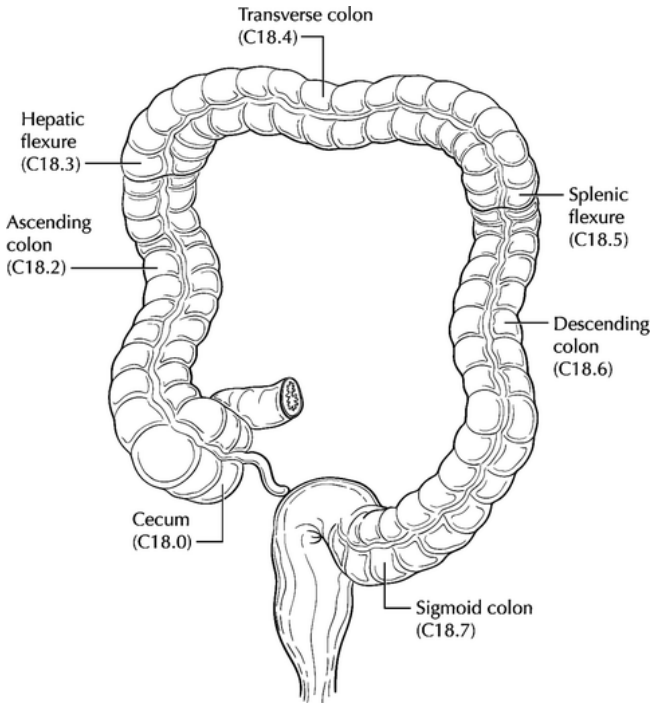


Figure 5.3: Topography codes for specific sections of the colon (taken from [Compton et al., 2012]).

automatically. For the remaining subjects, this process will occur if the given subject is added to the subject handled by the coding assistant.

5.4 Use Case

This section shows how an operator could describe their question to the coding assistant and how a coding expert could review this question, using the illustrating example from the previous sections.

5.4.1 Asking a Question

An operator can ask questions using the corresponding form in the interface. Asking a question is done in two steps. In the first step, seen in figure 5.4, the operator has to specify the subjects of their question, the cancer types and the version of the applying coding standards. This specification is separated from the rest of the form because the subject influences which information will be required in the description of the patient record. The same form is going to be used to edit the case later on, which explains the presence of the workshop date. This form will typically be filled in by the registry staff after the case has been discussed in such a workshop.

The second step concerns the patient record. The description of the patient record consists mainly of three sections: general information on the patient (age, gender), summary information on past and present tumors and exam descriptions (type, results, etc.), as seen in figure 5.5. In the final section, the operator may add any additional information they deem useful, in a free text comment box.

Once both forms have been filled in and submitted, the new problem will be saved. The coding assistant will then notify coding experts that a new question has been asked, and if there are subjects which can be automatically answered, the system will attempt to do so. In the example, the topography is such a subject, thus the system will attempt to solve it using the method designed in this work. More details about the method itself can be found in chapter 6.

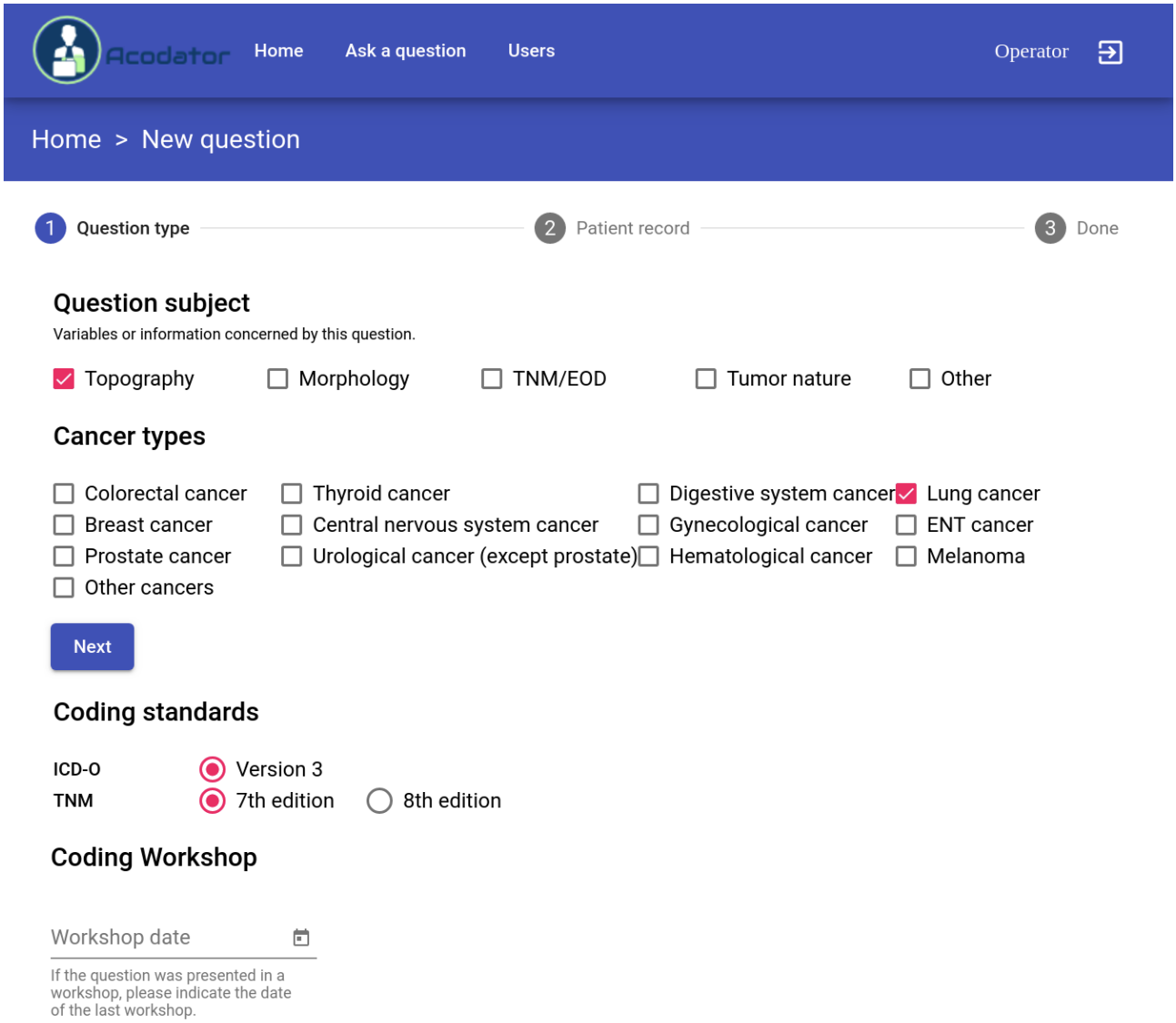


Figure 5.4: When asking a question, one must first indicate subject, cancer type and coding standard versions.

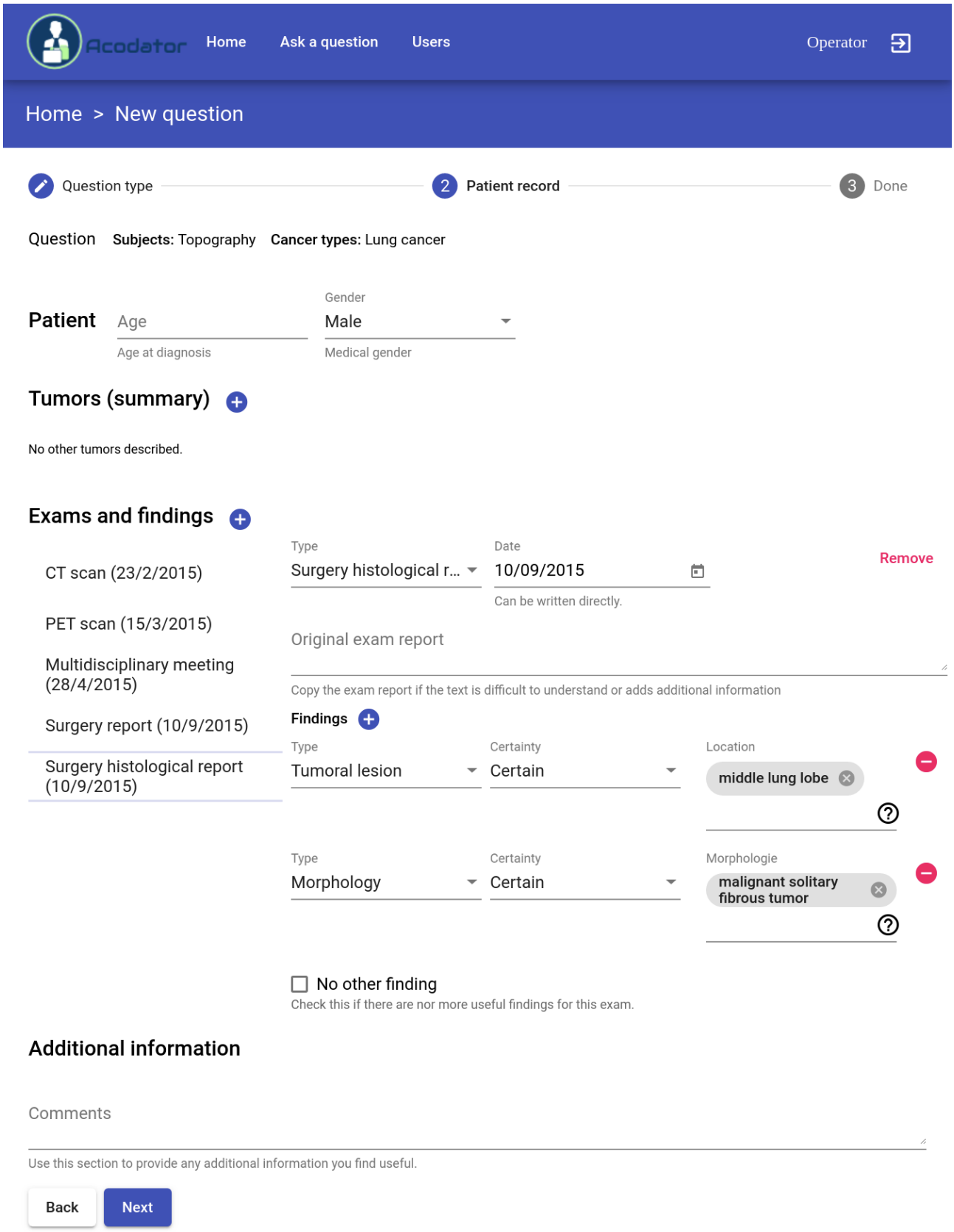


Figure 5.5: When asking a question, after providing basic information about the question, a description of the patient record must be given.

5.4.2 Reviewing a Case

Once a case has been solved, it can be reviewed by a coding expert. The interface available to experts can be seen in figure 5.6. The case review ultimately serves two purposes. First, to validate the provided answer, or, if there is no answer yet, to provide it. This step will be particularly crucial in the initial phases of the system, as it will still be learning how to answer most situations. Second, to prepare the case for future reuse, if the case is deemed interesting. Experts may then review the case, in order to remove unnecessary information and/or to generalize the described situation, so that future similar questions can best benefit from this new source case.

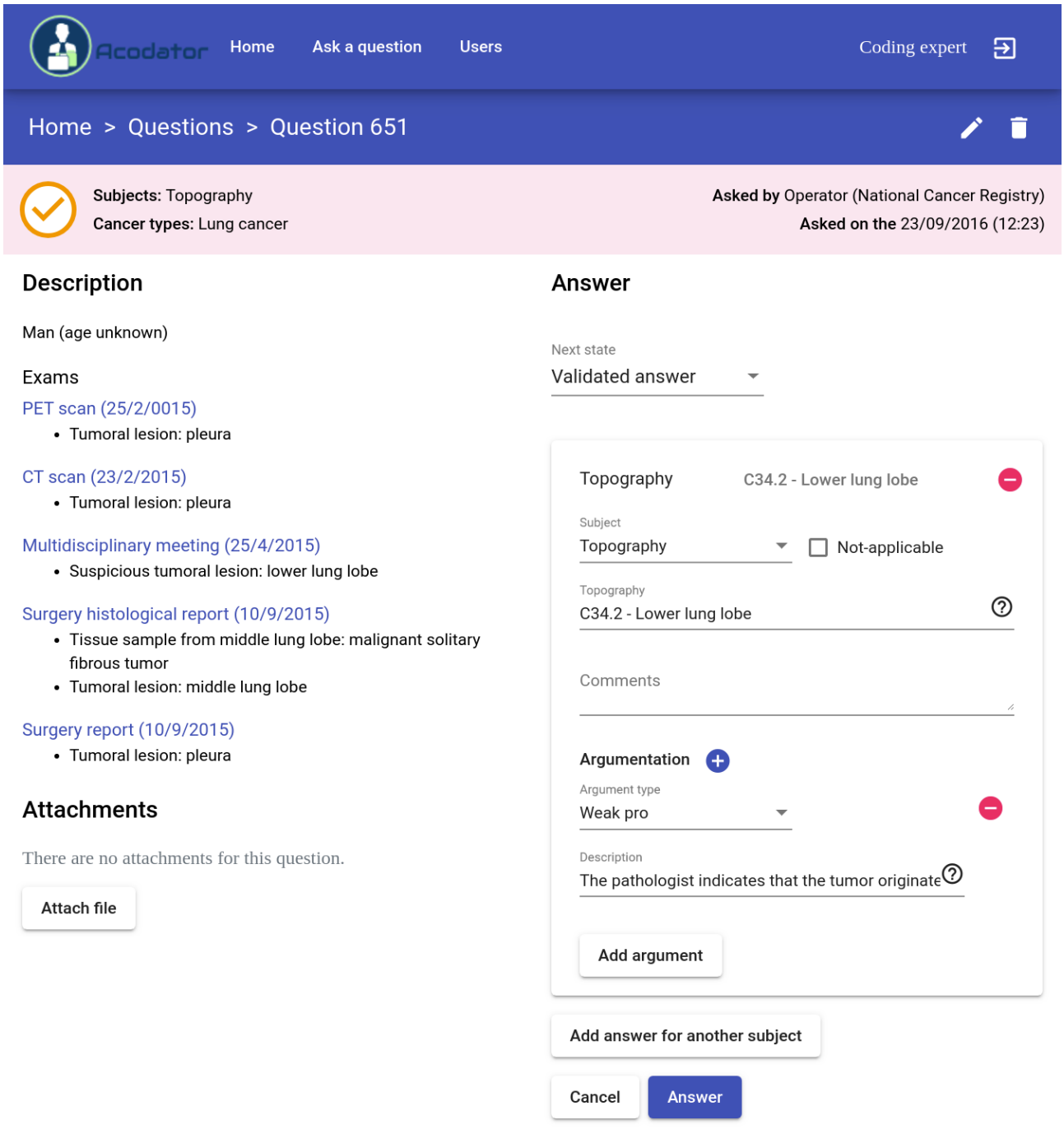


Figure 5.6: When reviewing a question, a coding expert can see the described patient record, attachments and the tentative solution provided by the coding assistant. They then may have to modify the answer and add or remove arguments.

Chapter 6

Case Retrieval and Reuse

To solve a problem in a case-based reasoning system based on a 4-R cycle, the first step is to retrieve a source case. In this step, previously solved problems are browsed to find a similar source case. To define what classifies two problems as being similar, the current solving process of the coding experts of the NCR has been analyzed. From this analysis, a method relying mainly on arguments has been designed, with a retrieval approach using arguments and a reuse approach building an explanation in the form of an argumentation for the answer. These approaches are described in the following sections.

6.1 Running Example

As a running example for this chapter, let us consider as the target problem the medical case of a male patient. The subject of the question for this target problem is the topography of the tumor to be coded. In 2016, during an imaging, a tumoral lesion is found in the right lower lung lobe. In a following biopsy, the tumor morphology is identified as adenocarcinoma without the presence of the TTF1 marker. A whole body PET scan does not reveal any additional tumoral lesions. This patient is discussed in a multidisciplinary team meeting and the clinicians conclude that this is a primary lung cancer and suggest a surgical treatment. The patient undergoes surgery and in the histological report of the surgery, it is confirmed that the whole tumor has been removed.

The case base for this example contains five source cases.

The first source case $\texttt{srce}_1$ concerns the medical case of a male patient. The subject of the question for this target problem is the topography of the tumor to be coded. During an imaging, a suspicious tumoral lesion is identified in the left lung. In a following biopsy, the tumor morphology is reported as melanoma. This type of cancer mainly develops on the skin, however, no lesion is found. A PET scan is performed but fails to locate any tumoral lesion except for the known pulmonary lesion. In a letter, the oncologist concludes that the pulmonary lesion is a metastasis of another tumor for which the original location is unknown. In a multidisciplinary team meeting, this patient is discussed and the oncologist's conclusion is confirmed. No treatment is performed as the patient dies soon afterwards.

For this particular problem, the chosen answer is `C80.9` (unknown location). This decision was taken because of the morphology of the tumor, i.e. melanoma, which is a cancer type which does not typically develop in the lungs. However, the lungs are a prime location for metastases. Both the oncologist and the multidisciplinary team meeting come to the same conclusion, i.e. that the tumor be found in the lungs is a metastasis of another tumor, which has not been found. This reasoning can be partially found in the argumentation containing four arguments, two weak pros ($\texttt{wp}_1^1$, $\texttt{wp}_1^2$) and two weak cons ($\texttt{wc}_1^1$, $\texttt{wc}_1^2$), stating

$\texttt{wp}_1^1$ An oncologist concludes that the primary location is unknown.

$\texttt{wp}_1^2$ A multidisciplinary team meeting concludes that the primary location is unknown.

$\texttt{wc}_1^1$ An imaging report indicates a tumoral lesion in the left lung.

$\texttt{wc}_1^2$ Except for a pulmonary tumoral lesion, no other lesions have been found.

The second source case $\texttt{srce}_2$ concerns the medical case of a female patient. The subject of the question is the topography. In 2013, an imaging report indicates a tumoral lesion in the upper left lung lobe. In a following biopsy, the tumor cell type is identified as adenocarcinoma with the presence of the TTF1 marker. A PET scan reveals no other tumoral lesions besides the known one in the lung. In a multidisciplinary team meeting, the patient is discussed and the clinicians conclude that this is a primary lung cancer and a surgical treatment is suggested.

For this particular problem, the chosen answer is `C34.1` (upper lung lobe) and the argumentation contains three weak pros ($\texttt{wp}_2^1$, $\texttt{wp}_2^2$, $\texttt{wp}_2^3$) stating

$\texttt{wp}_2^1$ An adenocarcinoma with the presence of the TTF1 marker is in favor of a primary lung cancer.

$\texttt{wp}_2^2$ A multidisciplinary team meeting concludes that this is a primary lung cancer.

$\texttt{wp}_2^3$ Except for a pulmonary tumoral lesion, no other lesions have been found.

It can be noted that $\texttt{wp}_2^3$ is the same argument as $\texttt{wc}_1^2$, but with a different argument type.

The third source case $\texttt{srce}_3$ concerns the medical case of a female patient. The subject of the question is the topography. In 2014, due to lingering chest pains, an imaging is performed and the report indicates a tumoral lesion in the right middle lung lobe. In a following biopsy, the tumor morphology is identified as non-Hodgkin lymphoma. In a letter, the treating oncologist indicates that the lung lesion is a non-Hodgkin lymphoma, and that no other lesion has been found. There is no further information available as the patient switched to another hospital afterwards.

For this particular problem, the chosen answer is `C34.2` (middle lung lobe). The argumentation contains three weak pros ($\texttt{wp}_3^1$, $\texttt{wp}_3^2$, $\texttt{wp}_3^3$) stating

$\texttt{wp}_3^1$ For lymphomas, if no lymph node or assimilated lymphatic structure is invaded by the tumor, the topography code associated to the invaded organ should be used.

$\texttt{wp}_3^2$ An imaging report indicates a tumoral lesion in the middle lung lobe.

$\texttt{wp}_3^3$ Except for a pulmonary tumoral lesion, no other lesions have been found.

The fourth source case $\texttt{srce}_4$ concerns the medical case of a female patient. The subject of the question is the topography. In 2014, because of abdominal pains, an imaging is done and a tumoral lesion in the ascending colon is found. A following PET scan shows additional lesions in the right lung lobe and in the liver. A biopsy in the colon allows the identification of the tumor as adenocarcinoma. In a letter, the treating oncologist concludes that the tumor originated in the colon and then spread to the lung and liver, i.e. that these tumors are metastases. Given the advanced stage of the disease, the patient is given palliative care.

For this particular problem, this chosen answer is `C18.2` (ascending colon). The argumentation contains six arguments, four weak pros ($\texttt{wp}_4^1$, $\texttt{wp}_4^2$, $\texttt{wp}_4^3$, $\texttt{wp}_4^4$) and two weak cons ($\texttt{wc}_4^1$, $\texttt{wc}_4^2$) stating

$\texttt{wp}_4^1$ An imaging report indicates a tumoral lesion in the ascending colon.

$\texttt{wp}_4^2$ A PET scan report indicates a tumoral lesion in the ascending colon.

$\texttt{wp}_4^3$ An oncologist concludes that the primary location is the ascending colon.

$\texttt{wp}_4^4$ An oncologist concludes that the lung and liver tumoral lesions are metastases of a primary colon tumor.

$\texttt{wc}_4^1$ A PET scan report indicates a tumoral lesion in the left lung.

$\texttt{wc}_4^2$ A PET scan report indicates a tumoral lesion in the liver.

It can be noted that for $\texttt{wp}_4^4$, this conclusion is confirmed by medical knowledge indicating that the liver and the lungs are prime locations for colon cancer metastases.

The fifth source case $\texttt{srce}_5$ concerns the medical case of a female patient. The subject of the question is the morphology. In a biopsy of a tumoral breast lesion, two morphologies are found, an in situ lobular carcinoma and an infiltrating ductal carcinoma.

For this particular problem, the chosen answer is `8522/3` (infiltrating duct and lobular carcinoma). The argumentation contains one strong pro ($\texttt{sp}_5^1$) stating

> $\texttt{sp}_5^1$ A breast tumor containing elements of a lobular carcinoma and a ductal carcinoma, with one of them being infiltrating, should be coded as `8522/3` (infiltrating duct and lobular carcinoma).

6.2 Retrieval

The designed coding assistant attempts to mimic the experts' solving process. This process is depicted in the next section. To achieve this, a retrieve method relying on arguments has been designed. In this context, an argument is defined as a specific part of the reasoning process of a coding expert when answering a question. This approach is quite different from the classical approach which would rely on the similarity between patient records to find similar problems. The goal of this approach is to find a way to answer questions and provide an explanation. This explanation is a crucial aspect of the coding assistant. It helps explain the chosen answer, and it also introduces a pedagogical aspect in the coding assistant. This allows the assistant to serve as a training tool for new operators, enabling them to start coding earlier.

6.2.1 Coding Expert Reasoning

When asking questions, the operators of the NCR detail their problem in a textual, sometimes semi-structured manner, usually in the form of a description of the relevant exams and findings, accompanied by answer propositions and/or original anonymized exam reports. With this description, the coding experts attempt to provide an answer and an explanation. After carefully reading the depiction, the experts check if the problem fits any of the situations described in the coding standards or coding manuals. If this is the case, then they solve the problem using the rules or guidelines found and reference them to explain their answer. If the applied standard or guideline is not clear for an operator, additional explanations may be provided. If none of the rules or guidelines apply, then the experts will look at previously answered questions, to verify if a similar problem can be found. To identify a similar problem, experts currently rely on their memory and then consult the documents of the various workshops to confirm their suspicion. While reviewing similar questions, the experts compare the subject of the question, the cancer type, and the difficulties faced by the operator. These difficulties include a hesitation between answers, comprehension issues concerning the information or terms contained in the reports, a difficulty to apply a coding rule or guideline or, missing or contradictory information. If the experts deem the problems to be similar, they have a closer look at the solving process used. If they find that it can still be applied to the new problem, they use it to answer the target problem. If the reasoning cannot be used, the existing one is adapted or a new one is provided. It is also possible to look up another previously solved question in an attempt to find a more suitable reasoning.

6.2.2 Argument Types

From the discussions with operators and coding experts, several argument types were identified. When discussing questions during workshops, several answers and their supporting arguments are presented, before a final answer is chosen. These supporting arguments can be strong or weak. A strong argument leaves little doubt as to the final answer of the question, even though most arguments are contextual.

In fact, for many situations, arguments can become invalid if new information is added. For weak arguments, the answer is not so certain. In those situations, a coding expert's choice is usually the decisive event for the chosen answer.

A pro is an argument which supports one answer or a set of answers. Let us consider an animal species classification task, where given some features about an animal, its species is to be identified. An example of a strong argument that supports only one answer, i.e. one species, is the argument stating that

If the animal has a beak, is a mammal and lays eggs, then it is a platypus.

If the given features are not enough to determine the exact species, a set of species can be listed. For example, the weak argument stating that

If it has wings and it is a mammal, then it is one of the species of the order Chiroptera.

supports any answer of bat species. A con is an argument which is against an answer, either by supporting an answer which is incompatible with the reference answer or because it directly defeats the reference answer. For example, given the problem of identifying the species of an ostrich, the weak argument stating that

If it cannot fly, then it is not a species of the Aves (bird) class.

is an argument against any answer of bird species. This argument directly defeats the given answer. The weak argument stating that

If it has wings and it cannot fly, then it is a species of the Spheniscidae family.

supports a set of answers (e.g. penguin species) which does not contain the given answer to the target problem.

The two features described above, i.e. for or against and strong or weak, lead to the creation of three types of arguments, namely

strong pros: strong arguments in favor of the answer to the question,
weak pros: weak arguments in favor of the answer *and*
weak cons: weak arguments against the answer.

For a given argument, the type can vary from question to question, as the definition of the type depends on the answer that is given. This change concerns the pro/con aspect in particular. The weak/strong feature changes more rarely, as it relies on domain knowledge rather than the given answer.

It was decided to exclude strong cons. In fact, using the definition of our features, a strong con would be an argument which leaves no doubt that the given answer is wrong. While this could be interesting pieces of knowledge to exclude answers or to validate a given answer, they are not useful for our argumentation method.

Let us consider the medical case of a patient for which a tumoral lesion is found in the left lung and for which a histological analysis reveals that the tumor is an adenocarcinoma and that the TTF1 cell marker is found. The question concerns the topography of the tumor to be coded. In the medical literature, it has been shown that often for a primary lung adenocarcinoma, the TTF1 marker is present. Thus the following argument can be made

When the TTF1 marker is present for an adenocarcinoma in the lungs, this tumor is likely a primary lung cancer.

If there is a tumoral lesion in the lungs, with an adenocarcinoma morphology and with the presence of the TTF1 marker, then this argument applies. Given its uncertain nature, it cannot be a strong argument. In fact, it is possible to have a primary adenocarcinoma with the presence of the TTF1

```
ASK {
  # Tumoral lesion in the lungs
  ?exam1 finding [
      findingType tumoralLesion ;
      certainty certain ;
      location ?location
    ] .
  ?location rdfs:subClassOf snmifre:T-28000_S2 .
  # Adenocarcinoma morphology
  ?exam2 finding [
      findingType morphology ;
      certainty certain ;
      morphology ?morph
    ] .
  ?morph rdfs:subClassOf snmifre:M-81400_S3 .
  # TTF1 marker
  ?exam3 finding [
      findingType ttf1Marker ;
      certainty certain ;
      present yes
    ] .
}
```

Figure 6.1: SPARQL query associated with the argument stating *When the TTF1 marker is present for an adenocarcinoma in the lungs, this tumor is likely a primary lung cancer.*

marker with a topography outside of the lungs. Hence it is a weak argument. If the answer to a topography question is a lung location, then this argument is a weak pro. If the answer is not a lung location, then this argument is a weak con. In both cases, the formal representation of the argument is the same. It is shown in figure 6.1.

At this stage, the type of an argument is defined by the coding experts when the argument is being used in an argumentation. This typing cannot be done automatically, as the needed knowledge is not formally represented, as described in section 5.2. If this knowledge were to be available, then it could be used to determine the argument type and also to enrich an argumentation with additional arguments.

6.2.3 Comparing Source Cases

As mentioned before, the choice of the source case relies mainly on arguments. To identify the retrieved case, the source cases from the case base are ranked with regards to their suitability for solving the target problem. This ranking relies on a preorder $\preccurlyeq_{tgt}$ that uses argument applicability. This ranking is detailed in the following sections.

A case is defined as a pair ($\mathtt{pb}$, $\mathtt{sol}(\mathtt{pb})$), where $\mathtt{pb}$ is a problem description and $\mathtt{sol}(\mathtt{pb})$ is the solution given to $\mathtt{pb}$. A solution is composed of an answer and of three sets of arguments for strong pros, weak pros and weak cons.

The functions $\mathtt{sp}$, $\mathtt{wp}$, $\mathtt{wc}$ are used to obtain the strong pros, weak pros and weak cons of a source case. They are functions which take a source case and return the set of arguments of the matching type. For example, $\mathtt{sp}(\mathtt{srce}_1) = \varnothing$, $\mathtt{wp}(\mathtt{srce}_1) = \{\ \mathtt{wp}_1^1, \mathtt{wp}_1^2\ \}$ and $\mathtt{wc}(\mathtt{srce}_1) = \{\ \mathtt{wc}_1^1, \mathtt{wc}_1^2\ \}$.

An argument is applicable for a given problem if it is true in the context of the associated patient record. For example, an argument stating that

$\mathcal{N}^{\texttt{argt}}_{\texttt{tgt}}(\texttt{srce}_i)$		$\texttt{srce}_i$				
	i	1	2	3	4	5
	sp	0	0	0	0	0
argt	wp	0	2	1	0	0
	wc	1	0	0	0	0

Table 6.1: Summary of the number of applicable arguments for each of the source cases for the running example which are applicable to tgt.

An imaging report indicates a tumoral lesion in the middle lung lobe.

is true for a problem if in the patient record there is an imaging report with a finding of a tumoral lesion in the middle lung lobe. Formally an argument is represented by a function which takes a problem from the problem space and returns a boolean. An argument arg is applicable for a problem pb if and only if $\texttt{arg}(\texttt{pb}) = \texttt{TRUE}$.

All the arguments of a source case apply to the problem of the source case. For instance, for the source case $\texttt{srce}_1$ of the running example, there are four arguments in the argumentation, namely $\texttt{wp}^1_1$, $\texttt{wp}^2_1$, $\texttt{wc}^1_1$ and $\texttt{wc}^2_1$. All four of these arguments apply to the problem of $\texttt{srce}_1$, meaning that $\texttt{wp}^1_1(\texttt{srce}_1) = \texttt{TRUE}$, $\texttt{wp}^2_1(\texttt{srce}_1) = \texttt{TRUE}$, $\texttt{wc}^1_1(\texttt{srce}_1) = \texttt{TRUE}$ and $\texttt{wc}^2_1(\texttt{srce}_1) = \texttt{TRUE}$.

In order to assess the applicability of a reasoning, a simple approach consists of counting the applicable arguments. However, only the arguments for which a formal representation, i.e. a SPARQL ASK query, has been provided can be used in the method provided by this work. The remaining arguments can only be assigned to an argumentation by coding experts.

Let $\mathcal{N}^{\texttt{argt}}_{\texttt{tgt}}$ be a function parametrized by an argument type function $\texttt{argt} \in \{\texttt{sp}, \texttt{wp}, \texttt{wc}\}$ and a problem tgt which takes a source case srce and returns the number of arguments of type argt from the source case srce which are applicable to the problem tgt.

$$\mathcal{N}^{\texttt{argt}}_{\texttt{tgt}}(\texttt{srce}) = \mid \{\texttt{arg} \in \texttt{argt}(\texttt{srce}) \mid \texttt{arg}(\texttt{tgt}) = \texttt{TRUE}\} \mid$$

In the running example, $\texttt{srce}_1$ has two weak pros and two weak cons. Neither of the weak pros apply to tgt, thus $\mathcal{N}^{\texttt{wp}}_{\texttt{tgt}}(\texttt{srce}_1) = 0$. For the weak cons, $\texttt{wc}^1_1$ does not apply to tgt and $\texttt{wc}^2_1$ applies to tgt, hence $\mathcal{N}^{\texttt{wc}}_{\texttt{tgt}}(\texttt{srce}_1) = 1$. In table 6.1, the result of the application of $\mathcal{N}^{\texttt{argt}}_{\texttt{tgt}}$ for each argument type and for each source case in the running example is shown.

A preorder $\preccurlyeq_{tgt}$ is used to compare source cases using mainly arguments. To achieve this ordering, three criteria are defined $\mathcal{C}_{\texttt{strong}}$, $\mathcal{C}_{\texttt{weak}}$ and $\mathcal{C}_{\texttt{dist}}$. Each of them considers different information from the source cases and the preorder $\preccurlyeq_{tgt}$ combines the results to produce a final ranking.

Strong Arguments with $\mathcal{C}_{\texttt{strong}}$

The first criterion $\mathcal{C}_{\texttt{strong}}$ relies only on strong arguments. The underlying idea is that given two source cases $\texttt{srce}_i$ and $\texttt{srce}_j$ where for $\texttt{srce}_i$ there is a strong argument for the answer that applies and for $\texttt{srce}_j$ there is no applicable strong argument, then $\texttt{srce}_i$ is more suitable to solve tgt than $\texttt{srce}_j$.

Let $\Delta^{\texttt{s}}_{\texttt{tgt}}$ be a function parametrized by a problem tgt which takes two source cases and returns the difference of the numbers of strong pros of the two source cases which are applicable to tgt. A positive difference indicates that the first source case is more suitable, a negative difference when the second source case is more suitable. If the difference is equal to zero, then both are equally suitable according to this criterion. $\Delta^{\texttt{s}}_{\texttt{tgt}}$ is defined as

$$\Delta^{\mathtt{s}}_{\mathtt{tgt}}(\mathtt{srce}_i, \mathtt{srce}_j) = \mathcal{N}^{\mathtt{sp}}_{\mathtt{tgt}}(\mathtt{srce}_i) - \mathcal{N}^{\mathtt{sp}}_{\mathtt{tgt}}(\mathtt{srce}_j)$$

For the running example, there are no strong arguments in our case base which apply to the target problem. Thus for all the source cases, this difference will be zero, meaning that with regards to the strong arguments, none of the source cases is preferred over the others.

Let us consider an example with source cases containing strong arguments. In this example, the target problem `tgt` concerns the coding of the morphology of a breast tumor where two morphologies have been found, a lobular carcinoma and a ductal carcinoma. Two source cases $\mathtt{srce}_a$ and $\mathtt{srce}_b$ are compared, where $\mathtt{srce}_a$ has one strong pro $\mathtt{sp}^1_a$ stating

> *A breast tumor containing elements of a lobular carcinoma and a ductal carcinoma, with one of them being infiltrating, should be coded as* ***8522/3*** *(infiltrating duct and lobular carcinoma).*

and $\mathtt{srce}_b$ has one strong pro $\mathtt{sp}^1_b$ stating

> *A colon tumor containing elements of a villous adenocarcinoma and any other type of adenocarcinoma should be coded as* ***8262/3*** *(villous adenocarcinoma).*

In this example, $\mathtt{sp}^1_a$ applies to the target problem and $\mathtt{sp}^1_b$ does not. Thus

$$\Delta^{\mathtt{s}}_{\mathtt{tgt}}(\mathtt{srce}_a, \mathtt{srce}_b) = \mathcal{N}^{\mathtt{sp}}_{\mathtt{tgt}}(\mathtt{srce}_a) - \mathcal{N}^{\mathtt{sp}}_{\mathtt{tgt}}(\mathtt{srce}_b) = 1$$

meaning that $\mathtt{srce}_a$ is preferred over $\mathtt{srce}_b$ to solve the target problem.

In real cases, there should not be many strong arguments. This is due to their nature, as they indicate that an answer is well-known. It is expected that these arguments will mostly appear for easier cases and are more likely to be of use for novice operators.

Weak Arguments with $\mathcal{C}_{\mathtt{weak}}$

The second criterion $\mathcal{C}_{\mathtt{weak}}$ relies only on weak arguments. Unlike the previous criterion, there are two types of arguments to consider, weak pros and weak cons. The underlying idea is still to identify the source case for which the reasoning can be best applied to the target problem. However, when it comes to not applicable arguments, cons do not impact the ranking in the same way as pros. In fact, a con which cannot be applied has a positive impact whereas a pro which cannot be applied has a negative impact. Thus, in this criterion $\mathcal{C}_{\mathtt{weak}}$, source cases with more applicable pros and fewer applicable cons should be considered more suitable. Similarly to the previous criterion, the number of applicable arguments is counted, and then transformed into a score using a weighted sum.

For the construction of the score, first the argumentations were reviewed. The weak arguments represent parts of the reasoning process used by the coding experts to answer a question. In this process, all arguments are not equally important. Some of them can be dropped or changed without impacting the final answer. This difference in importance can be represented using weights. However, it is difficult to define these weights. Coding experts have to be consulted for this task and it can be complicated to properly judge the difference in importance. Thus, for this initial approach, it was decided to give the same importance to all weak arguments. This impacts how to decide which source case is preferred to solve the target problem.

The underlying idea in the method designed in this work is to rely on argumentation to find similar problems. Thus, if the argumentation of a source case applies to the target problem, it is a good candidate for the retrieved case. If all of the weak arguments apply, then it seems reasonable to assume that this source case can be reused to solve the target problem. The situation is more complicated when some of the arguments do not apply. As weak cons represent reservations that the coding experts had regarding the chosen answer, if some of these cons do not apply, it strengthens the

$\Delta^{\mathtt{w}}_{\mathtt{tgt}}(\mathtt{srce}_i, \mathtt{srce}_j)$	j					Comments
i	1	2	3	4	5	
1		-8	-5	-2	-2	$\mathtt{srce}_1$ is less suited than all other source cases
2	8		3	6	6	$\mathtt{srce}_2$ is more suited than all other source cases
3	5	-3		3	3	$\mathtt{srce}_3$ is more suited than $\mathtt{srce}_1$, $\mathtt{srce}_4$ and $\mathtt{srce}_5$
4	2	-6	-3		0	$\mathtt{srce}_4$ is more suited than $\mathtt{srce}_1$
5	2	-6	-3	0		$\mathtt{srce}_5$ is more suited than $\mathtt{srce}_1$

Table 6.2: Result of the comparison with regards to $\mathcal{C}_{\mathtt{weak}}$ for all of the source cases of the running example.

answer of the considered source case. In a similar fashion, weak pros encourage the chosen answer. Thus, if some of these pros do not apply, it weakens the reasoning for the chosen answer.

The score built to compare source cases using weak arguments should incorporate these ideas. Similarly to the previous criterion, the number of applicable weak arguments is counted. However, a weight is associated to each argument. There are two weights at this stage, one for the weak pros and another for the weak cons. With the number of applicable arguments and their weights, a weighted sum is built, representing how fit a source case is for solving the target problem. The higher the score, the more interesting the source case is for solving the target problem.

Let $\Delta^{\mathtt{w}}_{\mathtt{tgt}}$ be a function parametrized by a target problem `tgt` which takes two source cases and returns an integer which indicates which source case is more suitable. A positive integer is used when the first source case is more suitable, a negative integer when the second source case is more suitable and 0 when both are equally suitable according to this criterion. $\Delta^{\mathtt{w}}_{\mathtt{tgt}}$ is defined as

$$\Delta^{\mathtt{w}}_{\mathtt{tgt}}(\mathtt{srce}_i, \mathtt{srce}_j) = \lambda_p \cdot (\mathcal{N}^{\mathtt{wp}}_{\mathtt{tgt}}(\mathtt{srce}_i) - \mathcal{N}^{\mathtt{wp}}_{\mathtt{tgt}}(\mathtt{srce}_j)) - \lambda_c \cdot (\mathcal{N}^{\mathtt{wc}}_{\mathtt{tgt}}(\mathtt{srce}_i) - \mathcal{N}^{\mathtt{wc}}_{\mathtt{tgt}}(\mathtt{srce}_j))$$

where λ_p and λ_c are nonnegative coefficients used to weight the importance of the weak pros with regards to the weak cons. Currently, these global parameters are set to $\lambda_p = 3$ and $\lambda_c = 2$, in order to focus more on the difference of applicable pros rather than on the difference of applicable cons. When more source cases are available, it could be interesting to review and optimize the value of these parameters λ_p and λ_c.

For the running example, the result of the application of $\Delta^{\mathtt{w}}_{\mathtt{tgt}}$ to all of the source cases is shown in table 6.2. Considering $\mathcal{C}_{\mathtt{weak}}$, $\mathtt{srce}_2$ is the most suitable source case, followed by $\mathtt{srce}_3$, $\mathtt{srce}_4$, $\mathtt{srce}_5$ and finally $\mathtt{srce}_1$.

Patient Records with $\mathcal{C}_{\mathtt{dist}}$

For the third and final criterion, the patient records of the compared source cases are considered. The underlying idea of this criterion is that if the difference of the patient record of $\mathtt{srce}_i$ to the patient record of `tgt` is smaller than the difference of the patient record of $\mathtt{srce}_j$ to the patient record of `tgt`, then $\mathtt{srce}_i$ is more suitable than $\mathtt{srce}_j$. This last criterion has been added as a default criterion to enable a ranking of source cases even if all the criteria based on arguments fail to provide a ranking.

The focus of the work of this project is on argumentation and its use for answering coding questions. However, it is possible to have source cases for which there are no formalized arguments. This can happen if the arguments have not yet been formalized, or if the formal representation of the argument is too complicated for the current approach. In this situation, the criterion $\mathcal{C}_{\mathtt{dist}}$ is used to determine if the target problem is close to the problem associated to the source case. This enables the use of those source cases and makes it easier for coding experts to find similar questions.

As patient records are represented using RDFS graphs, a graph edit distance is used to determine the distance between two patient records. As defined in section 4.4, the edit distance from the graph `source` to the graph `target` is defined as the sum of the cost of the edit operations of the best edit path.

In order to further facilitate the computation of the edit distance, the adjustments described in section 4.4 are used:

- consider only nodes for edit operations,
- consider graphs as trees, with the patient serving as root *and*
- substitute only nodes of the same type.

Three edit operations are used, namely:

- insert: adds a node from the target graph in the source graph;
- delete: removes a node from the source graph;
- substitute: replaces a node from the source graph with a node from the target graph.

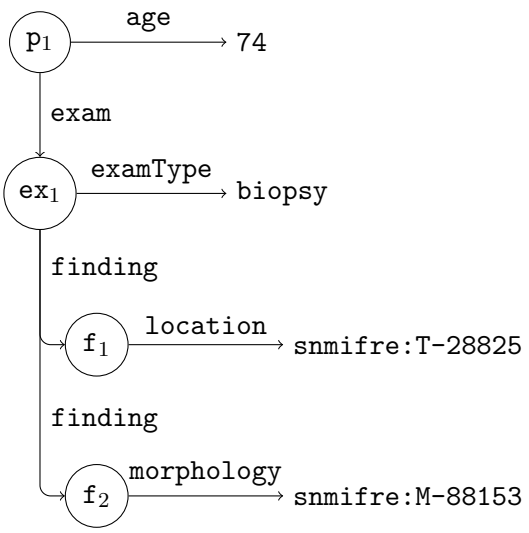
Both insert and delete operations have a fixed cost τ, which is a nonnegative parameter. Currently τ is set to 200. The substitute operation depends on the used nodes. The cost was built to be relatively small compared to τ, to encourage edit paths with more substitutions and fewer insertions or deletions. The substitute cost is calculated as follows:

- For blank nodes, the cost is 0 as they serve as generic containers for other nodes for our project.
- For the nodes of type `age`, which contain the age of a patient, the absolute value of the difference divided by 100 is used. This division is performed in order to normalize the difference and to reflect the minor importance of the age of the patient for the patient record similarity.
- For nodes that represent free text, like comments, the difference is ignored. In the beginning of this project, it was decided against analyzing free text information, to limit the complexity of this project.
- For nodes that represent location and morphology, a hierarchical distance is used to compute the cost. The hierarchical distance is described in more detail in section 4.4.
- For nodes that represent dates, e.g. exam date, the difference is ignored, hence the cost is 0. In fact, it is almost impossible to have two problems with the same dates for two different patients. So far, it has proven difficult to determine how dates could be used to compare patient records.
- For all the remaining types of nodes, if the labels of the compared nodes are the same then the cost is 0, otherwise the cost is 1.

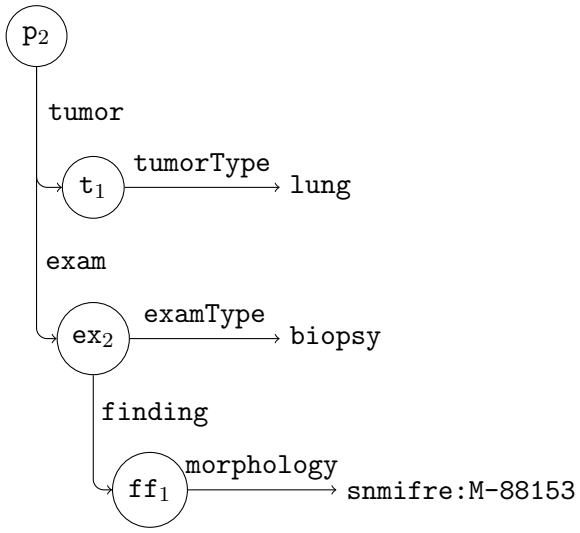
To compare dates, one interesting idea could be to rewrite dates with regards to a reference date for each patient record, and compare the rewritten dates. For example, the incidence date could be used as reference and the difference in dates could rely on the number of days since the incidence date. For example, given a patient record where the incidence date of the considered tumor is July 4th, 2014 and with one exam performed on the July 30th, 2014 and another patient record where the incidence date of the considered tumor is October 14th, 2015 and with one exam performed on the October 23rd, 2015, the dates of the exams could be compared with this approach. For the first exam, the date could be rewritten to `26`, as there are 26 days between the date of the exam and the incidence date. Similarly, for the second exam, the date could be rewritten to `9`. The difference could then be the absolute value of the difference between these two numbers. In this example, the difference would be `17`. Using this approach, it may be possible to focus on the relative time at which the exams have been performed and their order, rather than the absolute time.

As an illustrating example of an edit path and the edit distance, let us consider the graphs shown in figure 6.2. To transform the source graph into the target graph, the following edit path can be used:

- substitute $\mathtt{p}_1$ with $\mathtt{p}_2$
- delete `74` (patient age)
- insert $\mathtt{t}_1$



(a) Source graph.



(b) Target graph.

Figure 6.2: Example graphs used to illustrate the edit path between two graphs. Blank nodes are marked by a circle.

- insert `lung` (tumor type)
- substitute $\texttt{ex}_1$ with $\texttt{ex}_2$
- substitute $\texttt{f}_2$ with $\texttt{ff}_1$
- delete $\texttt{f}_1$
- delete `snmifre:T-28825` (location)

The total cost of this edit path is $5\tau = 1000$.

Let $\Delta^{\texttt{d}}_{\texttt{tgt}}$ be a function parametrized by a target problem `tgt` which takes two sources cases and returns a number which indicates which source case is more suitable. A positive or zero value indicates that the first source case is preferred and a negative number indicates that the second source case is preferred. Let `dist` be a function which takes two problems and returns the edit distance from the patient record of the first problem to the patient record of the second problem. Given two source cases $\texttt{srce}_i = (\texttt{pb}_i,\ \texttt{sol}(\texttt{pb}_i))$ and $\texttt{srce}_j = (\texttt{pb}_j,\ \texttt{sol}(\texttt{pb}_j))$, $\Delta^{\texttt{d}}_{\texttt{tgt}}$ is defined as

$$\Delta^{\texttt{d}}_{\texttt{tgt}}(\texttt{srce}_i, \texttt{srce}_j) = \texttt{dist}(\texttt{tgt}, \texttt{pb}_j) - \texttt{dist}(\texttt{tgt}, \texttt{pb}_i)$$

Ranking Source Cases with $\preccurlyeq_{tgt}$

In order to compare two source cases $\texttt{srce}_i$ and $\texttt{srce}_j$, a preorder $\preccurlyeq_{tgt}$ is used. It combines the results of the three criteria $\mathcal{C}_{\texttt{strong}}$, $\mathcal{C}_{\texttt{weak}}$ and $\mathcal{C}_{\texttt{dist}}$ described previously, considering them in a lexicographical order. Given a target problem `tgt`, $\texttt{srce}_i$ is considered more suitable to solve `tgt` than $\texttt{srce}_j$, i.e. $\texttt{srce}_i \preccurlyeq_{tgt} \texttt{srce}_j$ if

$$\begin{aligned} &\Delta^{\texttt{s}}_{\texttt{tgt}}(\texttt{srce}_i, \texttt{srce}_j) > 0 \\ \texttt{or}\ (&\Delta^{\texttt{s}}_{\texttt{tgt}}(\texttt{srce}_i, \texttt{srce}_j) = 0\ \texttt{and}\ (\quad \Delta^{\texttt{w}}_{\texttt{tgt}}(\texttt{srce}_i, \texttt{srce}_j) > 0 \\ &\qquad\qquad \texttt{or}\ (\Delta^{\texttt{w}}_{\texttt{tgt}}(\texttt{srce}_i, \texttt{srce}_j) = 0\ \texttt{and}\ \Delta^{\texttt{d}}_{\texttt{tgt}}(\texttt{srce}_i, \texttt{srce}_j) \geq 0))) \end{aligned}$$

In order to obtain the retrieved case for the running example, first all of the source cases with the same subject as the target problem are extracted from the case base. In this situation, there are four source cases to consider, $\texttt{srce}_1$, $\texttt{srce}_2$, $\texttt{srce}_3$ and $\texttt{srce}_4$. $\texttt{srce}_5$ is excluded as it concerns the

morphology. The retrieved case is the source case which is the most suited to solve $\texttt{tgt}$, using $\preccurlyeq_{tgt}$ to determine this ranking. In this example, the source cases are ranked as follows:

$$\texttt{srce}_2 \preccurlyeq_{tgt} \texttt{srce}_3 \preccurlyeq_{tgt} \texttt{srce}_4 \preccurlyeq_{tgt} \texttt{srce}_1$$

Thus, $\texttt{srce}_2$ is the retrieved case.

6.3 Reuse

Once a suitable source case has been found, this retrieved case is used to determine a possible solution for the target problem. As for the retrieve step, the approach of the coding experts of the NCR has been reviewed in order to design the reuse method. For this step, an important factor to consider is the evolution of the coding standards. Regularly, the standards are updated and the changes need to be applied to the coding decisions made in the past.

6.3.1 Reuse by Copy

As consistency is a very important requirement for cancer registries, it has been decided to adopt a reuse by copy approach for this project. In order to build the solution for the target problem, the answer of the retrieved case is copied. For the argumentation, all the arguments from the argumentation of the retrieved case that are applicable to the target problem are copied.

For the running example, the answer of the retrieved case is `C34.1` (upper lung lobe). For the argumentation, both $\texttt{wp}_2^2$ and $\texttt{wp}_2^3$ are applicable to $\texttt{tgt}$ and there are no strong pros or weak cons.

In the future, it could be of interest to analyze if more arguments could be reused for the argumentation of the target problem. In fact, in the case base, there may be other source cases with the same answer as the one chosen for the target problem. It could be of interest to add all arguments from these source cases which are applicable to the target problem to the new argumentation, even if the sources cases have not been used as the retrieved case. This would allow the method to provide a richer argumentation, though at the risk of including arguments which do not make sense. These additional arguments would have to be reviewed by the coding experts.

6.3.2 New Coding Standards

As the medical research progresses, the focus of the collected data may change, resulting in changes in the coding standards and coding practices. New information may need to be considered or more detailed information may need to be coded. For example, in 2016 the 8th edition of TNM tumor staging standard was released and it entered into effect in 2018 for the NCR. While the changes do not impact all of the codes and rules, for those impacted, there might be source cases which rely on these old rules. In this kind of situation, there are two possible options in order to be able to continue to use these source cases. Coding experts can review these source cases and, if necessary, update the answer or arguments. This may take a lot of time for coding experts and thus is a poor option. Another option consists in trying to integrate the changes into the domain knowledge and modify the reuse method to take into account this new knowledge. During the reuse step, this new knowledge can then be used. If the solution of the retrieved case relies on an outdated coding standard, instead of reusing the retrieved case in its current form, it can first be updated to follow the new version of the coding standard. This updated retrieved case makes it possible to provide a solution which complies with the new standard, even though its solution was provided in a different context. This option is preferred, as it does not so heavily rely on coding experts and it can also provide an explanation for the answer by indicating which changes were made to the original answer.

6.4 Use case

For the running example of this chapter, figure 6.3 shows how a solved problem and the retrieved case are displayed in the implemented coding assistant. The steps involved in asking a new question have already been shown previously and are not repeated in this section. At this stage, there is no highlighting of similarities between the target problem and the retrieved case in the coding assistant, nor is there any highlighting of the information used by a given argument. Both of these features could be added to facilitate the understanding of explanations by operators.

6.5 Conclusion

Medical coding for a cancer registry is a difficult task. Operators and coding experts are facing many challenges. Over time, solutions have been implemented to help cope with these difficulties. In this chapter, a method has been presented to assist in the coding for a cancer registry. This new method attempts to use arguments, as they have shown their usefulness in the past for the NCR. In that context, arguments have been used to explain answers for operators and coding experts. The method proposed in this project now uses arguments in conjunction with case-based reasoning to answer the coding questions to the operators of the NCR.

Arguments are used in an innovative manner in the retrieve step of the case-based reasoning implementation designed in this project. Using three criteria $\mathcal{C}_{\texttt{strong}}$, $\mathcal{C}_{\texttt{weak}}$ and $\mathcal{C}_{\texttt{dist}}$, the source cases in the case base are ranked. This allows the identification of the retrieved case, i.e. the source case with the most suitable argumentation to solve the target problem.

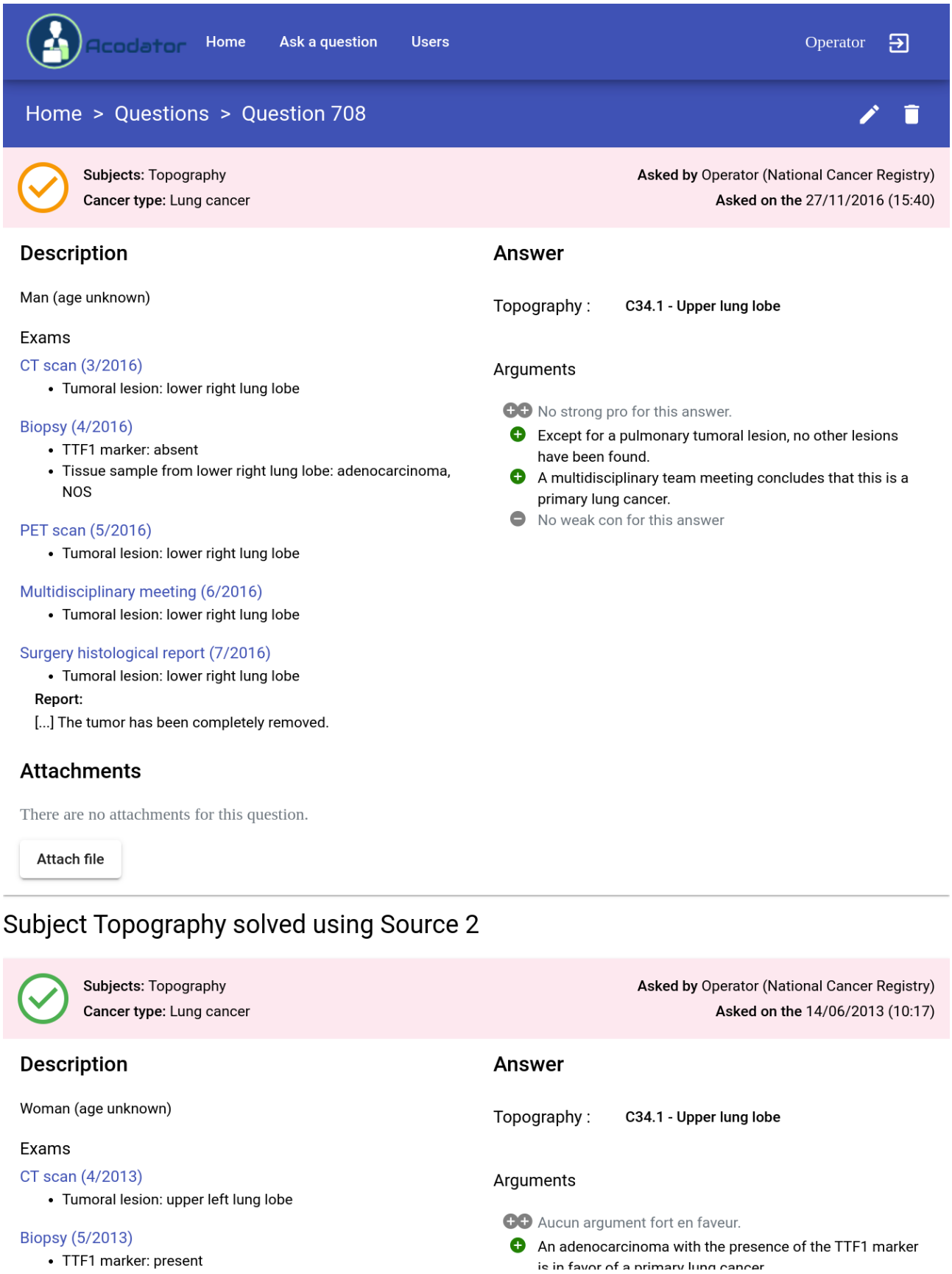


Figure 6.3: When viewing a solved question, if a source case has been used to provide a solution, it is shown below the viewed question.

Chapter 7

Evaluation

While the described method may seem promising, an evaluation is needed to assess its actual performance. There are multiple aspects which are worth evaluating, like the correctness of the solutions, the user acceptance and trust, the time gain or loss due to the new coding assistant or the quality of the coded data. For the latter, the coding assistant is only one of many factors and thus any evaluation requires a more global approach than is possible in the context of this project. User acceptance and trust can be evaluated using a user study. However, the developed coding assistant is currently only deployed internally for the team of the NCR and not yet available for operators, thus any evaluation involving users is not possible. The usefulness and quality of the explanations also falls into the category of aspects which require users and cannot yet be tested. Thus, in a first step, only the correctness of the provided solutions is evaluated.

7.1 Method

In order to test the correctness of the solutions, there are two main possibilities. If there is a way to validate the correctness or optimality of a solution, it is sufficient to generate problems and to solve them. These problems can be artificially generated or taken from existing problem descriptions if these are available. However, for the medical coding problem faced in this project, there is no way to automatically validate the correctness of a solution. In fact, the validation process relies entirely on coding experts, i.e. a manual operation. Thus in order to evaluate the designed method, it is necessary to use a manually curated dataset.

7.1.1 Evaluation Set

The method used to collect the source cases is the same as the one described in section 5.3.1. In fact, the dataset will also be used by the coding assistant once it is made available to operators. These source cases are based on real questions and solutions of the NCR operators and have been validated by the coding experts of the NCR. The collected dataset contains only source cases related to topography questions. This choice is motivated by the availability of previously solved questions and the relative ease of the problem descriptions, as described in section 5.3.1. The collection and formalization of the previously asked questions and their solution is an ongoing process. At the moment of the evaluation, 38 source cases are available and all of these are used in the performance assessment. The dataset covers some of the questions asked between 2015 and 2016. For the patient records, there is an average of about 3 exams per patient, with an average of about 1.3 findings per exam. Table 7.1 summarizes the frequency of the encountered exam types and finding types. For the solutions, the answers cover 28 of the 333 possible topography codes, with 6 codes being used in at least two answers. Table 7.2 shows the used topography codes and their frequency. For the collected arguments, there are 71 in the dataset with 61 that have a formal representation, i.e. a SPARQL ASK query which can be used by the coding assistant. Table 7.3 shows the use of the various arguments and their type. On average, the argumentation in the solution is composed of 2 arguments. An argument can be used in more

Exam Type	#
Biopsy	32
CT scan	25
Opinion clinician	20
Histological surgery report	18
Multidisciplinary team meeting	15
PET scan	13
MRI	10
Other	10
Other Imaging	6
Ultrasound	3
Bronchoscopy	2
Surgery report	2
Blood test	1
Colonoscopy	1
Mammography	1
Total	159

Finding Type	#
Tumoral lesion	76
Morphology	47
Adenopathy	13
Peritoneal Carcinomatosis	9
TTF1 marker	3
Abscess of skin	1
β-HCG	1
Multiple pulmonary opacities	1
Total	151

Table 7.1: Exam type and finding type distribution in the evaluation dataset.

than one argumentation, however, on average, an argument is used only in 1 argumentation.

7.1.2 Indicators

For the evaluation, two setups have been tested. The first setup aims at testing whether the designed approach can solve problems that are present in the case base. This test is important to validate the consistency of the approach. This setup uses all the source cases in the evaluation dataset. For each source case `srce` = (`pb`, `sol`(`pb`)), the associated problem `pb` is given to the coding assistant. The provided solution is compared with the original solution `sol`(`pb`) from the dataset. It is expected to be identical and the solution should be computed using the original source case `srce`. The number of correct answers is counted and used for the assessment. Similarly, the number of arguments and the arguments used in the provided solutions are observed and compared to the expected argumentation.

The goal of the second setup is to evaluate how well the designed method can solve new, unseen problems. For the second setup, a leave-one-out cross-validation is used. This is necessary because of the small dataset available for the evaluation. Given a larger dataset, the source cases could have been split into a case base and an evaluation base. In each iteration of the cross-validation, one source case `srce` = (`pb`, `sol`(`pb`)) is selected. The associated problem `pb` is given to the coding assistant to be solved using as case base all the case base except for the selected source case `srce`. The number of correct answers is counted and used for the assessment. The provided arguments are observed and compared with the expected one, like in the other setup. This test is split in two parts. The first part uses all of the dataset, whereas the second only considers the source cases for which there is at least one other source case with the same answer.

7.2 Results

For the first setup using all of the curated dataset, of the 38 problems tested, 34 have been correctly answered. For 5 problems, the retrieved case is different from the expected retrieved case. For the argumentation provided with each solution, in 24 only the expected arguments are present. In 2

Topography code	#	Topography code	#
C00.0 - External upper lip	1	C44.9 - Skin, NOS	1
C05.0 - Hard palate	1	C48.2 - Peritoneum, NOS	4
C08.9 - Major salivary gland, NOS	1	C50.5 - Lower-outer quadrant of breast	1
C11.0 - Superior wall of nasopharynx	1	C51.9 - Vulva, NOS	1
C14.0 - Histological surgery report	1	C54.1 - Endometrium	1
C20.9 - Pharynx, NOS	1	C56.9 - Ovary	2
C21.1 - Anal canal	1	C57.9 - Female genital tract, NOS	1
C34.0 - Main bronchus	3	C60.9 - Penis, NOS	1
C34.1 - Upper lobe, lung	1	C67.9 - Bladder, NOS	2
C37.9 - Thymus	1	C69.6 - Orbit, NOS	1
C38.0 - Heart	1	C71.8 - Overlapping lesion of brain	2
C41.2 - Bone marrow	1	C71.9 - Brain, NOS	1
C44.0 - Skin of lip, NOS	3	C77.8 - Lymph nodes of multiple regions	2
C44.6 - Skin of upper limb and shoulder	2	C80.9 - Unknown primary site	4

Table 7.2: Topography code coverage in evaluation dataset.

	Strong pro	**Weak pro**	**Weak con**	**All**
Total number	8	58	12	78
Average per case	0.2	1.5	0.3	2.1
Range per case	0–1	0–4	0–3	0–5

Table 7.3: Argument use in the evaluation set by argument type.

Id	Exp.	Prov.	Other answers
1	C77.8	C77.8	C48.2 C14.0 C67.9 C54.1
2	C80.9	C80.9	C80.9 C34.0 C56.9 C77.8
3	C34.0	C34.0	C34.0 C71.8 C48.2 C08.9
4	C56.9	C56.9	C48.2 C48.2 C57.9 C48.2
5	C34.1	C34.1	C34.0 C80.9 C14.0 C48.2
6	C08.9	C08.9	C71.8 C05.0 C48.2 C77.8
7	C44.6	C44.6	C50.5 C71.8 C80.9 C08.9
8	C51.9	C51.9	C60.9 C00.0 C67.9 C44.9
9	C14.0	C14.0	C48.2 C71.9 C56.9 C67.9
10	C05.0	C08.9	C05.0 C80.9 C71.8 C77.8
11	C54.1	C54.1	C11.0 C56.9 C80.9 C20.9
12	C80.9	C80.9	C56.9 C05.0 C11.0 C54.1
13	C71.8	C71.8	C08.9 C34.0 C77.8 C48.2
14	C20.9	C20.9	C08.9 C56.9 C11.0 C48.2
15	C21.1	C21.1	C20.9 C67.9 C44.9 C44.0
16	C60.9	C60.9	C00.0 C51.9 C44.9 C67.9
17	C11.0	C08.9	C11.0 C54.1 C56.9 C80.9
18	C44.9	C44.9	C00.0 C60.9 C21.1 C51.9
19	C37.9	C37.9	C77.8 C57.9 C71.8 C08.9
20	C44.0	C44.0	C67.9 C60.9 C21.1 C00.0
21	C69.6	C69.6	C44.9 C67.9 C00.0 C21.1
22	C41.2	C41.2	C56.9 C44.6 C48.2 C50.5
23	C48.2	C48.2	C48.2 C57.9 C56.9 C48.2
24	C38.0	C38.0	C80.9 C80.9 C71.8 C54.1
25	C80.9	C80.9	C80.9 C80.9 C48.2 C71.8
26	C00.0	C00.0	C60.9 C51.9 C44.9 C67.9
27	C50.5	C50.5	C71.8 C77.8 C80.9 C08.9
28	C56.9	C57.9	C56.9 C11.0 C54.1 C80.9
29	C71.9	C71.9	C67.9 C14.0 C21.1 C48.2
30	C67.9	C67.9	C67.9 C21.1 C71.9 C44.0
31	C48.2	C48.2	C48.2 C48.2 C14.0 C71.9
32	C48.2	C48.2	C48.2 C48.2 C48.2 C71.8
33	C48.2	C48.2	C48.2 C48.2 C71.8 C34.0
34	C77.8	C56.9	C77.8 C37.9 C71.8 C05.0
35	C34.0	C34.0	C34.0 C80.9 C14.0 C48.2
36	C80.9	C80.9	C38.0 C80.9 C11.0 C54.1
37	C67.9	C67.9	C67.9 C51.9 C44.0 C60.9
38	C57.9	C57.9	C56.9 C37.9 C71.8 C77.8

Table 7.4: Detailed results for the provided answers in the first evaluation setup. First column identifies the problem, the second column shows the expected answer and the following five columns show the answer for the five closest source cases (first column being closest). The third column contains the provided answer. Correct answers are underlined.

solutions, there are more arguments than expected and it can be noted that for these two solutions the answer is wrong. For the remaining 12 solutions, there are fewer arguments and for two of these solutions the answer is wrong. For the 4 problems for which the provided solution is wrong, the expected retrieved case is present among the top five closest source cases, typically in second position. The detailed results can be found in table 7.4 and table 7.5.

For the second setup, 10 of the problems tested are properly answered. Of the remaining 28, for 2 the correct answer can be found in one of the top five closest source cases. For the provided argumentation, 4 solutions contain exactly the expected arguments. For 2 solutions, there are more arguments than expected and for 13 solutions there are fewer arguments. For 1 solution, the number of arguments is the same, but the arguments are different. For the remaining 18 solutions, there are no arguments at all. The detailed results can be found in table 7.6 and table 7.7.

7.3 Discussion

In the first setup, almost all the source cases of the evaluation dataset are correctly solved. For those which are not, a source case with the correct answer can still be found in the top five closest cases, indicating that there is little missing in order to solve these problems correctly. A closer review of these source cases shows that a major issue at this stage is the small amount of arguments and of formalized arguments for the source cases. In fact, given the strong dependence on arguments of the method designed in this project, it seems reasonable that a lack of arguments is detrimental to the performance. Increasing the number of arguments per source case seems to be the best option to improve the provided solutions. The source cases in the dataset have been coded with the help of coding experts, nevertheless a second review of the argumentation has been carried out. It can be seen that the arguments often focus on a very specific aspect and leave out some more basic information or context. For coding exerts and expert operators, this is not an issue as these are trivial for them and do not need to be reminded. For the coding assistant and for novice operators however, this could be very important. Thus it could be interesting to add these arguments.

Id	Expected	Provided
1	wp 1*,2* wc 3*	wp 1*,2* wc 3*
2	wp 7*,8* wc 6*,9*	wp 8*
3	wp 58*,59*,60*	wp 58*,59*,60*
4	wp 13*,15*,62*,63* wc 61*	wp 13*,15*,62*,63* wc 61*
5	wp 18*,19* wc 20*	wp 18*,19* wc 20*
6	wp 53*,54*	wp 53*,54*
7	wp 35*,36	wp 35*
8	sp 48*	sp 48*
9	wp 64*,65*	wp 64*,65*
10	wp 68*	wp 53*,54*
11		
12	sp 51	
13	sp 81*	sp 81*
14	wp 69*,70*	wp 69*,70*
15	sp 55*	sp 55*
16	sp 56*	sp 56*
17	wp 83*,84* wc 85*,86*,87	wp 53*
18	sp 73*	sp 73*
19	wp 75*,76	wp 75*
20	wp 39*,40*,41*	wp 39*,40*,41*
21	wp 77*	wp 77*
22	wp 89*,90* wc 91*	wp 89*,90* wc 91*
23	wp 92*,93	wp 92*
24	wp 94*,95* wc 96*	wp 94*,95* wc 96*
25	wp 8*,105	wp 8*
26	sp 79*	sp 79*
27	sp 101*	sp 101*
28	wp 103* wc 104*	wp 103*
29	wp 80	
30	wp 45*,108*	wp 45*,108*
31	wp 109*	wp 109*
32	wp 111*,112*	wp 111*,112*
33	wp 109*,113*	wp 109*,113*
34		wp 63*
35	wp 10*,115*	wp 10*,115*
36	wp 116,117*,118*	wp 117*,118*
37	wp 45*,108*	wp 45*,108*
38	wp 13*,103*,119 wc 120	wp 13*

Table 7.5: Detailed results for the provided arguments in the first evaluation setup. First column identifies the problem, the next column shows the expected arguments and the last column shows the provided arguments. Arguments are identified using an id in this table and are marked with a * if they have a formal representation.

Id	Exp.	Prov.	Other answers
1	C77.8	C48.2	C14.0 C67.9 C54.1 C34.0
2	C80.9	C80.9	C34.0 C56.9 C77.8 C34.1
3	C34.0	C34.0	C71.8 C48.2 C08.9 C77.8
4	C56.9	C48.2	C48.2 C57.9 C48.2 C80.9
5	C34.1	C34.0	C80.9 C14.0 C48.2 C71.9
6	C08.9	C71.8	C05.0 C48.2 C77.8 C34.0
7	C44.6	C50.5	C71.8 C80.9 C08.9 C34.0
8	C51.9	C60.9	C00.0 C67.9 C44.9 C44.0
9	C14.0	C48.2	C71.9 C56.9 C67.9 C21.1
10	C05.0	C08.9	C80.9 C71.8 C77.8 C56.9
11	C54.1	C11.0	C56.9 C80.9 C20.9 C60.9
12	C80.9	C56.9	C05.0 C11.0 C54.1 C14.0
13	C71.8	C08.9	C34.0 C77.8 C48.2 C05.0
14	C20.9	C08.9	C56.9 C11.0 C48.2 C54.1
15	C21.1	C20.9	C67.9 C44.9 C44.0 C71.9
16	C60.9	C00.0	C51.9 C44.9 C67.9 C44.0
17	C11.0	C08.9	C54.1 C56.9 C80.9 C20.9
18	C44.9	C00.0	C60.9 C21.1 C51.9 C67.9
19	C37.9	C77.8	C57.9 C71.8 C08.9 C48.2
20	C44.0	C67.9	C60.9 C21.1 C00.0 C51.9
21	C69.6	C44.9	C67.9 C00.0 C21.1 C60.9
22	C41.2	C56.9	C44.6 C48.2 C50.5 C80.9
23	C48.2	C48.2	C57.9 C56.9 C48.2 C41.2
24	C38.0	C80.9	C80.9 C71.8 C54.1 C11.0
25	C80.9	C80.9	C80.9 C48.2 C71.8 C50.5
26	C00.0	C60.9	C51.9 C44.9 C67.9 C44.0
27	C50.5	C71.8	C77.8 C80.9 C08.9 C05.0
28	C56.9	C57.9	C11.0 C54.1 C80.9 C14.0
29	C71.9	C67.9	C14.0 C21.1 C48.2 C44.0
30	C67.9	C67.9	C21.1 C71.9 C44.0 C44.9
31	C48.2	C48.2	C48.2 C14.0 C71.9 C21.1
32	C48.2	C48.2	C48.2 C48.2 C71.8 C57.9
33	C48.2	C48.2	C48.2 C71.8 C34.0 C08.9
34	C77.8	C56.9	C37.9 C71.8 C05.0 C08.9
35	C34.0	C34.0	C80.9 C14.0 C48.2 C71.9
36	C80.9	C38.0	C80.9 C11.0 C54.1 C48.2
37	C67.9	C67.9	C51.9 C44.0 C60.9 C00.0
38	C57.9	C56.9	C37.9 C71.8 C77.8 C48.2

Table 7.6: Detailed results for the provided answers in the second evaluation setup. First column identifies the problem, the second column shows the expected answer and the following five columns show the answer for the five closest source cases (first column being closest). The third column contains the provided answer. Correct answers are underlined.

Id	Expected	Provided
1	wp 1*,2* wc 3*	
2	wp 7*,8* wc 6*,9*	wp 8*
3	wp 58*,59*,60*	wp 10*
4	wp 13*,15*,62*,63* wc 61*	wp 109*
5	wp 18*,19* wc 20*	
6	wp 53*,54*	
7	wp 35*,36	
8	sp 48*	
9	wp 64*,65*	
10	wp 68*	wp 53*,54*
11		
12	sp 51	
13	sp 81*	
14	wp 69*,70*	wp 53*
15	sp 55*	wp 70*
16	sp 56*	
17	wp 83*,84* wc 85*,86*,87	wp 53*
18	sp 73*	
19	wp 75*,76	
20	wp 39*,40*,41*	
21	wp 77*	
22	wp 89*,90* wc 91*	wp 63*
23	wp 92*,93	wp 109*
24	wp 94*,95* wc 96*	
25	wp 8*,105	wp 117*
26	sp 79*	
27	sp 101*	
28	wp 103* wc 104*	wp 103*
29	wp 80	
30	wp 45*,108*	wp 45*,108*
31	wp 109*	wp 109*
32	wp 111*,112*	wp 109*
33	wp 109*,113*	wp 109*
34		wp 63*
35	wp 10*,115*	wp 60*
36	wp 116,117*,118*	
37	wp 45*,108*	wp 45*,108*
38	wp 13*,103*,119 wc 120	wp 13*

Table 7.7: Detailed results for the provided arguments in the second evaluation setup. First column identifies the problem, the next column shows the expected arguments and the last column shows the provided arguments. Arguments are identified using an id in this table and are marked with a * if they have a formal representation.

There are also some arguments which are difficult to formalize. When coding for a cancer registry, given the choice between a generic code (e.g. `C26.0` – intestinal tract, not otherwise specified) and a specific code (e.g. `C18.2` – ascending colon), the specific code should be preferred. However, it is not always possible to choose this code. There are situations where information is missing, and then a general code has to be used. In those situations, the argumentation for the provided answer may contain an argument which indicates this choice. The argument stating

There is no available information to determine the exact location of this colon cancer.

is such an example, supporting the use of the general intestinal tract topography code.

In this project, problem descriptions are given in the context of an open world assumption, meaning that it is not assumed that all the information is known. For example, if no tumoral lesion is reported in the described patient record, this does not imply that there is no tumoral lesion. It simply means that it is not known whether there is one. It is possible that the report which indicates a tumoral lesion has not been added to the problem description for various reasons. Those reasons can be very valid, e.g. for an exam report which is only available in a different hospital than the one in which the current operator is working for. In order to partially handle some of these situations, a feature is defined in the exam part of the patient record description for exams where the operator can confirm that there is no more information available.

For the second setup, the previous remarks concerning the arguments remain valid. In this test, 10 out of 38 problems are correctly solved. This seemingly low score can for the most part be explained by the small size of the available dataset. In fact, for 22 source cases, given that the correct answer does not appear in any other source case, the coding assistant in its current state cannot solve these problems. This is because of the reuse by copy approach and the fact that solutions contain a topography code as answer. Considering only the source cases for which there is at least one other source case with the expected answer, 10 out of 16 questions were correctly solved.

To improve the number of correctly solved problems, it might be interesting to change the reuse approach. For topography questions, this could be achieved by formalizing the transformation of the

body parts in the findings of the various exams into the topography code used in the answer. This approach could leverage existing mappings between SNOMED body parts and ICD-O topography codes. For example, from the mappings it is known that the body part ascending colon is mapped to the topography code `C18.2`. Given a question for which exam findings point to the fact that the tumor started in the ascending colon, the coding assistant could leverage this knowledge to determine that the final answer should be `C18.2`.

As for the argumentation, for most source cases, there are fewer arguments in the provided solutions and the arguments are different from the expected ones. Given that for many arguments, they are only used in one solution, it is normal that the original argumentation cannot be rebuilt by the system. This limitation can be overcome by adding more source cases and reusing existing arguments whenever possible. However, this could also be an indication that the existing arguments are inadequate for the intended use in this project. Some of them might be too specific to be easily reused without modification.

7.4 Conclusion

In this chapter, a preliminary evaluation of the method designed in this project has been presented. Despite the small dataset, this method has shown its ability to solve similar problems and provide a tentative argumentation. However, it has also shown that solutions, in particular arguments, need to be carefully formalized in order to obtain correct answers and good explanations.

Chapter 8

Conclusion and Future Work

In this project, the challenge of medical coding is faced. In particular, this work focuses on the coding for longitudinal studies, e.g. registries. There are many different types of registries, covering different topics and cancer registries are just one example. These different registries often share similar needs. This is particularly the case for cancer registries, as a collaborative effort has been made to obtain a clear and common definition of their goals and requirements.

To follow up on a global scale on the trends in cancer and to assess in the most efficient way public health policies and progress in oncology research, it is necessary to have high quality, reliable and comparable data. To achieve this requirement, it is crucial for data to be coded in a correct, efficient and consistent way. The definition of cancer registries also led to the development of rules and guidelines. Some resulted from the same collaboration and are shared and followed by all cancer registries. Those are the international coding standards, created by organizations like the International Association of Cancer Registries (IACR) or the European Network of Cancer Registries (ENCR) for European registries.

Despite their complex nature, the coding standards do not cover all possible situations encountered by operators. Lacking central guidelines, each cancer registry has developed their own approach to solve those situations. For the NCR, several measures have been implemented:

- operators may ask questions to the coding experts of the NCR
- difficult coding questions can be handled by the coding committee of the NCR
- coding questions are presented and discussed in the monthly coding workshops of the NCR

This project has been started in an attempt to reduce the time burden of these measures. The application aim of this work is the development of a coding assistant. The scientific aim is to study how case-based reasoning can be used to assist a person tasked with extracting information from multiple data sources, where data can be missing or contradictory, following defined rules and guidelines. The solutions provided by the case-based reasoning system should be explainable, to render them more understandable for operators and coding experts.

8.1 Contributions

In order to achieve these goals, the current procedures of the NCR have been analyzed. From this analysis it has been decided to implement a coding assistant. During the course of this project, a first version of this tool has been developed, focusing on solving topography questions. This tool serves as an intermediate between operators and coding experts, attempting to answer coding questions. Operators can use it to ask partially structured questions, for which the coding assistant provides tentative solutions. Coding experts may then review and validate these solutions.

The scientific aim of this project is to analyze how case-based reasoning can be used to assist a user in a setting where they have to extract information from multiple data sources. During this project, such an approach has been proposed, leveraging arguments provided by domain experts to assist a user in their task. This method is capable of dealing with missing data as well as contradictory data.

By integrating newly solved problems, this method also manages to solve new problems and provide similar coding for similar situations, which is a strong requirement for the application domain of this project.

The developed coding assistant relies on the designed method to solve coding questions. It provides a mean for operators and coding experts to identify some of the similar situations they have encountered in the past. To find the retrieved case, first strong arguments, then weak arguments and finally patient record similarity are used. The provided answer is accompanied by the arguments which helped provide this answer and serve as an explanation. This method has undergone a preliminary evaluation, which highlighted some strengths and limitations.

This work is a first step into designing a method capable of answering coding questions. It resulted from an analysis of current problem solving methods, their strengths and limitations. It is important for the approach to be able to explain its solution for coding questions. Given that arguments play an important role in the discussions between operators and coding experts, it was decided to include them in the formal answering method. This effort has resulted in the method described in the previous chapters. This method attempts to reproduce the reasoning process of the coding experts, by leveraging small pieces of their reasoning. Those pieces are called arguments and they represent small bits of domain knowledge and expert decisions, needed to answer coding questions, but also for medical coding in general.

This project has provided some insights into how medical data and argumentation can be represented, in particular in the context of medical coding assistance. A first and relatively simple approach for representing patient records and arguments has been produced. This was partly achieved by the use of very similar representations for patient record and for arguments, making it very easy to test for the applicability of an argument.

To compare patient records, a similarity measure has also been designed. It has been adapted from current approaches on edit distances for graphs and for trees. The current approach can be applied to other tree structures and is not limited to the medical domain.

There has been other work on combining argumentation and case-based reasoning in the past. This design is sometimes referred to as interpretive case-based reasoning [Kolodner, 1992]. Applications focused a lot on the legal domain [Rissland et al., 2005], e.g. assisting lawyers in their argumentation [Ashley, 1991, Aleven and Ashley, 1997] or helping during negotiations [Sycara, 1990].

The main difference of this work compared to previous interpretive case-based reasoning resides in the use of arguments. In previous work, arguments are mostly part of the solution or they are used to justify or criticize the provided solution. The retrieval step in the case-based reasoning approaches does not consider arguments and relies on other techniques. The legal reasoning system HYPO [Ashley, 1991] aims at assisting lawyers in their argumentation. In a trial, to convince a jury of their case, a lawyer may present previous trials and their outcome. These previous trials serve as precedents and are meant to support the desired outcome. In their presentation, the lawyer highlights the features which are similar in both situations (current and precedent). This precedent and the similarities are part of the arguments presented by lawyers during trials. The aim of the HYPO system is to assist lawyers in finding those precedents and in identifying how to present the similarities between the retrieved case and the current situation. To find the retrieved case, HYPO uses selected features, called dimensions. Arguments are only used later to explain why the retrieved case is interesting to the viewpoint of the lawyer.

The method proposed in this research is not complete, and there are many possible prospects for improvements and future work.

8.2 Domain Knowledge

One of the identified problems is the small amount of formalized source cases. While case-based reasoning can function with few source cases, performance is expected to be better with a larger case base. Unfortunately it takes time and skills in medical knowledge engineering to formalize source cases, in particular coding expert time. An interesting avenue to facilitate the knowledge acquisition process could be to include more cancer registries, in particular their coding experts. The various cancer registries face similar coding issues and would all benefit from the knowledge provided by the coding experts collective, reducing the overall bottleneck on argument formalization. One major obstacle for this approach is of course the language used to interact with operators and coding experts. The arguments and problem description would need to be translated into the different languages, which might also be difficult to achieve. Relying on translators might provide a solution for this problem. The formalization started in this project is still ongoing and once the system is used routinely, it should take less time. Thus, even though the problem should therefore improve over time, there are possibilities which could already be explored to address this issue.

Apart from those unprocessed questions, there is still a lot of available knowledge which can be added to the coding assistant to increase the performance. The coding guides authored by the NCR for its operators are one such example. These guides focus on one cancer type, e.g. breast, lung or prostate, and summarize the most important tips and rules for operators for the main information to code, e.g. topography and morphology. They also present the relevant anatomical concepts, regional lymph node areas and other specific relevant information. They are created by combining the recurring coding questions and errors, in order to provide operators with a quick and easy to access document. The knowledge contained in these guides could be used to create new artificial source cases and/or arguments. $\texttt{srce}_5$ introduced in chapter 6 is such a source case. In the coding guide for breast cancer, it is described that if both in situ lobular carcinoma and infiltrating ductal carcinoma morphologies are found, then the morphology code `8522/3` (infiltrating duct and lobular carcinoma) should be used. This also explains the typing of the argument as a strong pro.

Other important knowledge that could be added concerns the coding standards, in particular the changes between versions of these standards. As described in section 6.3.2, source cases which use older versions of a coding standard could then continue to be reused without needing manual revision by coding experts. The cancer staging of a lung cancer using the TNM staging system can be used as an example. Starting from 2018, the 7^{th} edition is replaced by the 8^{th} edition. For lung tumor, this has introduced several changes. For the T category of the tumor staging, in the 7^{th} edition, the code $\texttt{T}_2$ concerns tumors with a size between 3 and 7 cm, while in the 8^{th} edition, this range has been changed to 3 to 5 cm. Another change concerns the $\texttt{T}_3$ code used for tumors which are larger than 7 cm. In the 8^{th} edition, these tumors are instead coded with $\texttt{T}_4$. These are relatively small changes, nevertheless they require new answers and updated arguments, with a new description and an adapted SPARQL ASK query. For the M category which describes metastases, in the 7^{th} edition, the code $\texttt{M}_{1b}$ is used for tumors which have at least one extrathoracic metastasis. In the 8^{th} edition, this code has been split in two codes $\texttt{M}_{1b}$ (exactly one extrathoracic metastasis) and $\texttt{M}_{1c}$ (more than one extrathoracic metastasis). In this situation, the argument stating that

> *If there is an extrathoracic metastasis, then the code* $\texttt{M}_{1b}$ *should be used.*

could be split into two arguments, one for each new code, stating

> *If there is exactly one extrathoracic metastasis, then the code* $\texttt{M}_{1b}$ *should be used.*

and

> *If there are at least two extrathoracic metastases, then the code* $\texttt{M}_{1c}$ *should be used.*

When solving a new coding question for cancer staging, the correct code should be used, depending on which argument is applicable for the target problem. The coding assistant should also be able to solve questions using both versions. There is ongoing research for this type of problem, in particular for updating annotations of medical concepts in medical documents, e.g. the Evolution of Semantic Annotations (ELISA) project[1] [Cardoso et al., 2018]. Combining this work with the explanations built on arguments could be an interesting topic for future work.

8.3 Case Representation

Adding more domain knowledge might impact the chosen case representation. At this stage, patient records in particular use a simplified modeling in order to facilitate problem description. However, it could be argued that, in some situations, important nuances are lost by simplifying the description. In Luxembourg, the structuring of the hospital records is a very recent endeavor. When this project started, most information was still only available in textual form, mostly electronically, with little structure and consistency across health practitioners. Only last year, hospitals started to use structured and electronic hospital records. Most processes are still being upgraded. Given this progress, it could be interesting to compare how data are described in these systems, in order to analyze if some of these information could be automatically extracted when asking questions. This work could also be reused by the NCR to feed these data directly into the registry, further reducing the coding workload of operators. By making the case representation more intricate, it could also be easier for operators to describe some of the relevant information. In fact, the closer the descriptions are, the less interpretation is needed to transcribe the content of the medical record.

Some changes could also be done for solutions, both for answers and for arguments. As mentioned in 7.3, currently, the answer contains a specific code or text. As an illustrating example, let us consider a source case with a patient record with one imaging report indicating a tumoral lesion in the upper lung lobe and some mediastinal adenopathies, i.e. that the tumor has spread to some mediastinal lymph nodes. Knowing that mediastinal adenopathies are regional adenopathies for the found tumor lesion and that the domain knowledge states that the upper lung lobe is coded with the topography code `C34.1`, the answer for the topography question would be `C34.1`. Let the target problem to solve be a problem where the patient record has one imaging report indicating a tumoral lesion in the ascending colon and several pericolic adenopathies, which are regional adenopathies for an ascending colon cancer. With the current approach, when reusing the presented source case, the coding assistant would answer the target problem with the topography code `C34.1`. This answer is obviously wrong, since there is no mention of any lung lesion. However, the two situations can be solved using a similar reasoning. For the target problem, the expected topography code is `C18.2` (ascending color). As for the source case, the regional adenopathies support the notion of a primary ascending colon cancer.

By changing the content of the solution of a source case, it might be possible to support this kind of reasoning. Currently for topography and morphology questions, the answer contains a specific code. Instead, it could be interesting to store how this specific code was computed from the domain knowledge and the patient record. For the illustrating example, the answer could be a formula to deduce the apropriate topography code. For this answer it would be necessary to know which adenopathies are regional for the different cancer locations and how body parts are mapped into topography codes. For example, mediastinal adenopathies are regional adenopathies of lung cancer and pericolic adenopathies are regional adenopathies of ascending colon cancer.

Of course, it is not always easy to identify which information should be used. In the previous example, there is only one relevant finding and thus finding the right one was trivial. For most real cases, this is not true.

[1] `https://www.elisa-project.lu`

8.4 Argumentation

There are several aspects of the arguments which could be improved, notably the argument types. The current argument types, while straightforward and easy to use, provide only a rough approximation of the reasoning process of the coding experts. By replacing the current strong/weak typing with fine tuned weights, which could be different from source case to source case, it could be possible to design a system which would build an explanation for a given answer in order to solve a problem. These new weights might be able to more accurately represent the nuanced importance of the various arguments for the reasoning of the coding experts than the weights introduced in the section 6.2.3. Given a larger amount of users, both operators and coding experts, it might be possible to make use of crowdsourcing [Brabham, 2013]. Each user could provide their version of the importance of the various arguments. These views could then be combined, giving more weight to the important arguments and as such increasing their relevance for the problems. This could also enable continuous updating of these weights by allowing users to revise their opinion. This approach or other techniques could be explored in future work.

In this work, coding expert reasoning is only partially represented. Given the various arguments in favor and against an answer, the decision-making process of the coding expert is not included into the current approach. This could be achieved by modeling the complete reasoning process. This modeling could take the form of a proof, where the final conclusion is the answer to the question, and arguments represent intermediate steps. This approach is more complex than the current one, but could potentially solve more problems and provide richer explanations. This could be achieved by allowing the system to adapt these proofs. Let us consider a situation where in the retrieved case, the proof for the answer relies on the fact that the topography for the tumor to be coded is `C34.1`. If the reasoning used to determine the topography code in the retrieved case cannot be reproduced for the target problem, it could be possible to find a different source case where a different reasoning was used to determine the same topography (`C34.1`). If such a source case is found, the proof from the retrieved case could be adapted by replacing the initial reasoning for the topography with the reasoning from the second source case. This improvement represents a very interesting avenue for future work.

8.5 Coding Assistant

For the implementation of the coding assistant, the next step would be to make it available to the operators and coding experts of the NCR. During this initial deployment, further evaluations will be performed to extend the results from the preliminary evaluation. This makes it possible to define the priorities for future work and to assess how viable this kind of solution is for other registries. In fact, there are many advantages to having a shared coding assistant across multiple registries. For example, case authoring can be split among the registries, allowing each to contribute new source cases and validate existing ones, which would reduce the overall burden of this task for coding experts. This could also allow the differences in coding to be shared and discussed. One challenge which might emerge is the handling of different coding decisions for the same problem. Should the registries be unable to agree on a common coding, the different decisions will have to be made by the assistant in order to allow a continued use for the conflicting registries. This issue could have a solution similar to that of the problem concerning different versions of the coding standards, however further research is necessary to test this idea. Another challenge arising from the sharing of a single coding assistant is the language used in source cases. While the interface and formal concepts might be more easily translated into the various languages needed, the textual description of arguments and other textual descriptions may prove more difficult to translate quickly and reliably. Automatic translation systems could be of interest to solve this issue. Data privacy issues might also need to be addressed, especially since exam reports are difficult to properly anonymize.

The implemented coding assistant will continue to be improved by the daily use at the NCR, and it will later be generalized for testing in another registry. Other disease registries or medical documentation would be the most likely candidates for a future extension. The developed assistant

might also be used for registries outside the medical domain. For instance, it could also be of interest for domains where the solution cannot easily be determined and where explanations are important for end users and possibly also domain experts.

There are still many interesting prospects to explore. This work has only been a first step into designing and implementing a coding assistant for medical coding, starting with the NCR.

Bibliography

[Aamodt and Plaza, 1994] Aamodt, A. and Plaza, E. (1994). Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI communications*, 7(1):39–59.

[Abdel-Aziz et al., 2013] Abdel-Aziz, A., Cheng, W., Strickert, M., and Hüllermeier, E. (2013). *Preference-Based CBR: A Search-Based Problem Solving Framework*, pages 1–14. Springer Berlin Heidelberg, Berlin, Heidelberg.

[Abdel-Aziz and Hüllermeier, 2015] Abdel-Aziz, A. and Hüllermeier, E. (2015). Case base maintenance in preference-based cbr. In *International Conference on Case-Based Reasoning*, pages 1–14. Springer.

[Abdel-Aziz et al., 2014] Abdel-Aziz, A., Strickert, M., and Hüllermeier, E. (2014). *Learning Solution Similarity in Preference-Based CBR*, pages 17–31. Springer International Publishing, Cham.

[Abidi and Manickam, 2002] Abidi, S. S. R. and Manickam, S. (2002). Leveraging xml-based electronic medical records to extract experiential clinical knowledge: An automated approach to generate cases for medical case-based reasoning systems. *International Journal of Medical Informatics*, 68(1-3):187–203.

[Adadi and Berrada, 2018] Adadi, A. and Berrada, M. (2018). Peeking inside the black-box: A survey on explainable artificial intelligence (xai). *IEEE Access*, 6:52138–52160.

[Aha et al., 2001] Aha, D. W., Breslow, L. A., and Muñoz-Avila, H. (2001). Conversational case-based reasoning. *Applied Intelligence*, 14(1):9–32.

[Aleven and Ashley, 1997] Aleven, V. and Ashley, K. D. (1997). Teaching case-based argumentation through a model and examples empirical evaluation of an intelligent learning environment. In *Artificial intelligence in education*, volume 39, pages 87–94.

[Aronson et al., 2007] Aronson, A. R., Bodenreider, O., Demner-Fushman, D., Fung, K. W., Lee, V. K., Mork, J. G., Névéol, A., Peters, L., and Rogers, W. J. (2007). From indexing the biomedical literature to coding clinical text: Experience with mti and machine learning approaches. In *Proceedings of the Workshop on BioNLP 2007: Biological, Translational, and Clinical Language Processing*, BioNLP '07, pages 105–112, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Arp et al., 2014] Arp, D., Spreitzenbarth, M., Hubner, M., Gascon, H., Rieck, K., and Siemens, C. (2014). Drebin: Effective and explainable detection of android malware in your pocket. In *Ndss*, volume 14, pages 23–26.

[Ashley, 1991] Ashley, K. D. (1991). *Modeling legal arguments: Reasoning with cases and hypotheticals*. MIT press.

[Atkins et al., 2019] Atkins, T. J., Etemad, E. J., and Rivoal, F. (2018 (last consultation: November 2019)). Css snapshot 2018.

[Beckett and Berners-Lee, 2011] Beckett, D. and Berners-Lee, T. (2011). Turtle - Terse RDF Triple Language, `https://www.w3.org/TeamSubmission/turtle/`, W3C recommendation, last consultation: July 2019.

[Berners-Lee, 2009] Berners-Lee, T. (2009). The semantic web.

[Berners-Lee et al., 2001] Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The semantic web. *Scientific american*, 284(5):28–37.

[Bichindaritz et al., 2015] Bichindaritz, I., Marling, C., and Montani, S. (2015). Case-based Reasoning in the Health Sciences. In *Workshop Proceedings of ICCBR*.

[Bizer et al., 2011] Bizer, C., Heath, T., and Berners-Lee, T. (2011). Linked data: The story so far. *Semantic services, interoperability and web applications: emerging concepts*, pages 205–227.

[Brabham, 2013] Brabham, D. C. (2013). *Crowdsourcing*. Mit Press.

[Branting et al., 2004] Branting, K., Lester, J., and Mott, B. (2004). *Dialogue Management for Conversational Case-Based Reasoning*, pages 77–90. Springer Berlin Heidelberg, Berlin, Heidelberg.

[Brickley and Guha, 2014] Brickley, D. and Guha, R. V. (2014). RDF Schema 1.1, `https://www.w3.org/TR/rdf-schema/`, W3C recommendation, last consultation: March 2017.

[Bridge et al., 2006] Bridge, D., GÖKer, M. H., McGinty, L., and Smyth, B. (2006). Case-based recommender systems. *The Knowledge Engineering Review*, 20(3):315–320.

[Bunke and Messmer, 1993] Bunke, H. and Messmer, B. T. (1993). Similarity measures for structured representations. In *European Workshop on Case-Based Reasoning*, pages 106–118. Springer.

[Bygrave, 2001] Bygrave, L. A. (2001). Minding the machine: Article 15 of the ec data protection directive and automated profiling. *Computer Law & Security Report*, 17:17.

[Caminada and Gabbay, 2009] Caminada, M. W. A. and Gabbay, D. M. (2009). A logical account of formal argumentation. *Studia Logica*, 93(2):109–145.

[Cardoso et al., 2018] Cardoso, S., Reynaud-Delaître, C., Da Silveira, M., Lin, Y.-C., Gross, A., Rahm, E., and Pruski, C. (2018). Evolving semantic annotations through multiple versions of controlled medical terminologies. *Health and Technology*, 8(5):361–376.

[Chen and Wilkinson, 1998] Chen, H. and Wilkinson, L. J. (1998). Case match reduction through the integration of rule-based and case-based reasoning procedures. *Aha and Daniels*, pages 33–38.

[CISMEF, 2015] CISMEF, L. (2015). Systematized Nomenclature of MEDicine, version française, `http://bioportal.lirmm.fr/ontologies/SNMIFRE`, last consultation: August 2019.

[Cojan and Lieber, 2009] Cojan, J. and Lieber, J. (2009). *Belief Merging-Based Case Combination*, pages 105–119. Springer Berlin Heidelberg, Berlin, Heidelberg.

[Cojan and Lieber, 2014] Cojan, J. and Lieber, J. (2014). *Applying belief revision to case-based reasoning*, pages 133–161. Springer.

[Compton et al., 2012] Compton, C. C., Byrd, D. R., Garcia-Aguilar, J., Kurtzman, S. H., Olawaiye, A., and Washington, M. K. (2012). *Colon and Rectum*, pages 185–201. Springer New York, New York, NY.

[Crammer et al., 2007] Crammer, K., Dredze, M., Ganchev, K., Talukdar, P. P., and Carroll, S. (2007). Automatic code assignment to medical text. In *Proceedings of the Workshop on BioNLP 2007: Biological, Translational, and Clinical Language Processing*, pages 129–136. Association for Computational Linguistics.

[Cunningham et al., 2003] Cunningham, P., Doyle, D., and Loughrey, J. (2003). An evaluation of the usefulness of case-based explanation. In *International Conference on Case-Based Reasoning*, pages 122–130. Springer.

[EuroStat, 2019] EuroStat (2019). Causes of death statistics `https://ec.europa.eu/eurostat/statistics-explained/index.php/Causes_of_death_statistics` *(Last consultation: November 2019)*.

[Evans-Romaine and Marling, 2003] Evans-Romaine, K. and Marling, C. (2003). Prescribing exercise regimens for cardiac and pulmonary disease patients with cbr. In *Workshop on CBR in the health sciences at 5th international conference on case-based reasoning (ICCBR-03)*, pages 45–62. Citeseer.

[Fielding et al., 1997] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., and Berners-Lee, T. (1997). Hypertext transfer protocol–http/1.1.

[Folleso et al., 2014] Folleso, S. L., Heimark, O., and Ekerholt, M. (2014). A system for conversational case-based reasoning in multiple-disease medical diagnosis.

[Forbus et al., 1995] Forbus, K. D., Gentner, D., and Law, K. (1995). Mac/fac: A model of similarity-based retrieval. *Cognitive science*, 19(2):141–205.

[Franco et al., 2013] Franco, R., Rocco, G., Marino, F. Z., Pirozzi, G., Normanno, N., Morabito, A., Sperlongano, P., Stiuso, P., Luce, A., Botti, G., et al. (2013). Anaplastic lymphoma kinase: a glimmer of hope in lung cancer treatment? *Expert review of anticancer therapy*, 13(4):407–420.

[Gedikli et al., 2014] Gedikli, F., Jannach, D., and Ge, M. (2014). How should i explain? a comparison of different explanation types for recommender systems. *International Journal of Human-Computer Studies*, 72(4):367–382.

[Giboney et al., 2015] Giboney, J. S., Brown, S. A., Lowry, P. B., and Nunamaker Jr, J. F. (2015). User acceptance of knowledge-based system recommendations: Explanations, arguments, and fit. *Decision Support Systems*, 72:1–10.

[Golding and Rosenbloom, 1991] Golding, A. R. and Rosenbloom, P. S. (1991). Improving rule-based systems through case-based reasoning. In *AAAI*, pages 22–27.

[Gomez-Uribe and Hunt, 2015] Gomez-Uribe, C. A. and Hunt, N. (2015). The netflix recommender system: Algorithms, business value, and innovation. *ACM Trans. Manage. Inf. Syst.*, 6(4):13:1–13:19.

[Gregor and Benbasat, 1999] Gregor, S. and Benbasat, I. (1999). Explanations from intelligent systems: Theoretical foundations and implications for practice. *MIS quarterly*, pages 497–530.

[Gu, 2006] Gu, M. (2006). Knowledge-intensive conversational case-based reasoning in software component retrieval.

[Gu and Aamodt, 2005] Gu, M. and Aamodt, A. (2005). A knowledge-intensive method for conversational cbr. In *International Conference on Case-Based Reasoning*, pages 296–311. Springer.

[Gu and Aamodt, 2006] Gu, M. and Aamodt, A. (2006). Dialog learning in conversational cbr. In *FLAIRS Conference*, pages 358–363.

[Gärdenfors, 2003] Gärdenfors, P. (2003). *Belief revision*, volume 29. Cambridge University Press.

[Gómez-Gauchía et al., 2006] Gómez-Gauchía, H., Díaz-Agudo, B., and González-Calero, P. (2006). *Ontology-Driven Development of Conversational CBR Systems*, pages 309–324. Springer Berlin Heidelberg, Berlin, Heidelberg.

[Harris and Seaborne, 2013] Harris, S. and Seaborne, A. (2013). SPARQL 1.1, `https://www.w3.org/TR/sparql11-query`, W3C recommendation.

[Holt et al., 2005] Holt, A., Bichindaritz, I., Schmidt, R., and Perner, P. (2005). Medical applications in case-based reasoning. *Knowledge Eng. Review*, 20:289–292.

[Hüllermeier and Cheng, 2013] Hüllermeier, E. and Cheng, W. (2013). Preference-based cbr: General ideas and basic principles. In *IJCAI*. Citeseer.

[Hüllermeier and Schlegel, 2011] Hüllermeier, E. and Schlegel, P. (2011). *Preference-based CBR: First steps toward a methodological framework*, pages 77–91. Springer.

[Jalali and Leake, 2012] Jalali, V. and Leake, D. (2012). *Custom Accessibility-Based CCBR Question Selection by Ongoing User Classification*, pages 196–210. Springer Berlin Heidelberg, Berlin, Heidelberg.

[Joseph et al., 2016] Joseph, J. W., Chiu, D. T., Nathanson, L. A., and Horng, S. (2016). A rule based algorithm to generate problem lists using emergency department medication reconciliation. *International Journal of Medical Informatics*.

[Karacapilidis et al., 1997] Karacapilidis, N., Trousse, B., and Papadias, D. (1997). Using case-based reasoning for argumentation with multiple viewpoints. In *International Conference on Case-Based Reasoning*, pages 541–552. Springer.

[Katsuno and Mendelzon, 1991] Katsuno, H. and Mendelzon, A. O. (1991). Propositional knowledge base revision and minimal change. *Artificial Intelligence*, 52(3):263–294.

[Kavuluru et al., 2013a] Kavuluru, R., Han, S., and Harris, D. (2013a). Unsupervised extraction of diagnosis codes from emrs using knowledge-based and extractive text summarization techniques. In *Canadian Conference on Artificial Intelligence*, pages 77–88. Springer Berlin Heidelberg.

[Kavuluru et al., 2013b] Kavuluru, R., Hands, I., Durbin, E. B., and Witt, L. (2013b). Automatic Extraction of ICD-O-3 Primary Sites from Cancer Pathology Reports. *AMIA Summits on Translational Science Proceedings*, 2013:112–116.

[Kavuluru et al., 2015] Kavuluru, R., Rios, A., and Lu, Y. (2015). An empirical evaluation of supervised learning approaches in assigning diagnosis codes to electronic medical records. *Artificial Intelligence in Medicine*, 65(2):155–166.

[Kolodner, 1992] Kolodner, J. L. (1992). An introduction to case-based reasoning. *Artificial intelligence review*, 6(1):3–34.

[Konieczny et al., 2004] Konieczny, S., Lang, J., and Marquis, P. (2004). Da2 merging operators. *Artificial Intelligence*, 157(1–2):49–79.

[Lecornu et al., 2009] Lecornu, L., Thillay, G., Le Guillou, C., Garreau, P.-J., Saliou, P., Jantzem, H., Puentes, J., and Cauvin, J. M. (2009). Referocod: a probabilistic method to medical coding support. In *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 3421–3424. IEEE.

[Levenshtein, 1966] Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710.

[Lieber et al., 2008] Lieber, J., d'Aquin, M., Badra, F., and Napoli, A. (2008). Modeling adaptation of breast cancer treatment decision protocols in the kasimir project. *Applied Intelligence*, 28(3):261–274.

[Linden et al., 2003] Linden, G., Smith, B., and York, J. (2003). Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing*, (1):76–80.

[Manning et al., 1999] Manning, C. D., Manning, C. D., and Schütze, H. (1999). *Foundations of statistical natural language processing*. MIT press.

[Marling et al., 2014] Marling, C., Montani, S., Bichindaritz, I., and Funk, P. (2014). Synergistic case-based reasoning in medical domains. *Expert system with applications*, 41(2):249–259.

[Maximini et al., 2003] Maximini, K., Maximini, R., and Bergmann, R. (2003). *An investigation of generalized cases*, pages 261–275. Springer.

[Mcsherry, 2001] Mcsherry, D. (2001). Interactive case-based reasoning in sequential diagnosis. *Applied Intelligence*, 14(1):65–76.

[McSherry, 2003a] McSherry, D. (2003a). Explanation in case-based reasoning: an evidential approach. In *Proceedings of the 8th UK Workshop on Case-Based Reasoning*, pages 47–55.

[McSherry, 2003b] McSherry, D. (2003b). Increasing dialogue efficiency in case-based reasoning without loss of solution quality. In *IJCAI*, pages 121–126.

[McSherry, 2004] McSherry, D. (2004). Explaining the Pros and Cons of Conclusions in CBR. In *European Conference on Case-Based Reasoning*, pages 317–330. Springer.

[McSherry, 2009] McSherry, D. (2009). *Conversational Case-Based Reasoning in Medical Classification and Diagnosis*, pages 116–125. Springer Berlin Heidelberg, Berlin, Heidelberg.

[McSherry, 2011] McSherry, D. (2011). Conversational case-based reasoning in medical decision making. *Artificial intelligence in medicine*, 52(2):59–66.

[McSherry, 2014] McSherry, D. (2014). *An Algorithm for Conversational Case-Based Reasoning in Classification Tasks*, pages 289–304. Springer International Publishing, Cham.

[Melacci et al., 2008] Melacci, S., Sarti, L., Maggini, M., and Bianchini, M. (2008). A neural network approach to similarity learning. In *IAPR Workshop on Artificial Neural Networks in Pattern Recognition*, pages 133–136. Springer.

[Noussa-Yao et al., 2015] Noussa-Yao, J., Boussadi, A., Richard, M., Heudes, D., and Degoulet, P. (2015). Using a snowflake data model and autocompletion to support diagnostic coding in acute care hospitals. *Studies in health technology and informatics*, 210:334–338.

[Nugent et al., 2009] Nugent, C., Doyle, D., and Cunningham, P. (2009). Gaining insight through case-based explanation. *Journal of Intelligent Information Systems*, 32(3):267–295.

[Olsson et al., 2014] Olsson, T., Gillblad, D., Funk, P., and Xiong, N. (2014). Case-based reasoning for explaining probabilistic machine learning. *International Journal of Computer Science and Information Technology*, 6(2):87.

[Ontañón et al., 2015] Ontañón, S., Plaza, E., and Zhu, J. (2015). *Argument-Based Case Revision in CBR for Story Generation*, pages 290–305. Springer International Publishing, Cham.

[Patrick et al., 2007] Patrick, J., Zhang, Y., and Wang, Y. (2007). Developing feature types for classifying clinical notes. In *Proceedings of the Workshop on BioNLP 2007: Biological, Translational, and Clinical Language Processing*, BioNLP '07, pages 191–192, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Peppas, 2008] Peppas, P. (2008). *Belief Revision*.

[Pestian et al., 2007] Pestian, J. P., Brew, C., Matykiewicz, P., Hovermale, D. J., Johnson, N., Cohen, K. B., and Duch, W. (2007). A shared task involving multi-label classification of clinical free text. In *Proceedings of the Workshop on BioNLP 2007: Biological, Translational, and Clinical Language Processing*, pages 97–104, 1572411. Association for Computational Linguistics.

[Pons et al., 2016] Pons, E., Braun, L. M., Hunink, M. M., and Kors, J. A. (2016). Natural language processing in radiology: a systematic review. *Radiology*, 279(2):329–343.

[Prentzas and Hatzilygeroudis, 2007] Prentzas, J. and Hatzilygeroudis, I. (2007). Categorizing approaches combining rule-based and case-based reasoning. *Expert Systems*, 24(2):97–122.

[Prentzas et al., 2008] Prentzas, J., Hatzilygeroudis, I., and Michail, O. (2008). Improving the accuracy of neuro-symbolic rules with case-based reasoning. In *1 st International Workshop on Combinations of Intelligent Methods and Applications (CIMA 2008)*, page 49.

[Reategui et al., 1997] Reategui, E. B., Campbell, J. A., and Leao, B. F. (1997). Combining a neural network with case-based reasoning in a diagnostic system. *Artificial Intelligence in Medicine*, 9(1):5–27.

[Resnick and Varian, 1997] Resnick, P. and Varian, H. R. (1997). Recommender systems. *Communications of the ACM*, 40(3):56–59.

[Richter and Weber, 2013] Richter, M. M. and Weber, R. O. (2013). *Case-based reasoning: a textbook*. Springer Science & Business Media.

[Riesbeck and Schank, 2013] Riesbeck, C. K. and Schank, R. C. (2013). *Inside case-based reasoning*. Psychology Press.

[Rissland et al., 2005] Rissland, E. L., Ashley, K. D., and Branting, L. K. (2005). Case-based reasoning and law. *The Knowledge Engineering Review*, 20(3):293–298.

[Rosenberg, 2014] Rosenberg, S. A. (2014). Decade in review—cancer immunotherapy: entering the mainstream of cancer treatment. *Nature reviews Clinical oncology*, 11(11):630.

[Rossille et al., 2005] Rossille, D., Laurent, J.-F., and Burgun, A. (2005). Modelling a decision-support system for oncology using rule-based and case-based reasoning methodologies. *International journal of medical informatics*, 74(2):299–306.

[Rudin, 2018] Rudin, C. (2018). Please stop explaining black box models for high stakes decisions. *arXiv preprint arXiv:1811.10154*.

[Saraiva et al., 2016] Saraiva, R., Perkusich, M., Silva, L., Almeida, H., Siebra, C., and Perkusich, A. (2016). Early diagnosis of gastrointestinal cancer by using case-based and rule-based reasoning. *Expert Systems with Applications*, 61:192–202.

[Saraiva et al., 2015] Saraiva, R. M., Bezerra, J., Perkusich, M., Almeida, H., and Siebra, C. (2015). A hybrid approach using case-based reasoning and rule-based reasoning to support cancer diagnosis: A pilot study. *Studies in health technology and informatics*, 216:862–866.

[Schnell et al., 2017] Schnell, M., Couffignal, S., Lieber, J., Saleh, S., and Jay, N. (2017). Case-Based Interpretation of Best Medical Coding Practices — Application to Data Collection for Cancer Registries. In *Conference Proceedings of ICCBR*.

[Shadbolt et al., 2006] Shadbolt, N., Berners-Lee, T., and Hall, W. (2006). The semantic web revisited. *IEEE intelligent systems*, 21(3):96–101.

[Shi et al., 2017] Shi, H., Xie, P., Hu, Z., Zhang, M., and Xing, E. P. (2017). Towards automated icd coding using deep learning. *arXiv preprint arXiv:1711.04075*.

[Smyth, 1998] Smyth, B. (1998). Case-base maintenance. In *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, pages 507–516. Springer.

[Smyth and Keane, 1995] Smyth, B. and Keane, M. T. (1995). Remembering to forget. In *Proceedings of the 14th international joint conference on Artificial intelligence*, pages 377–382. Citeseer.

[Stanfill et al., 2010] Stanfill, M. H., Williams, M., Fenton, S. H., Jenders, R. A., and Hersh, W. R. (2010). A systematic literature review of automated clinical coding and classification systems. *Journal of the American Medical Informatics Association*, 17(6):646–651.

[Stephens et al., 2009] Stephens, F. O., Aigner, K., Allen-Mersh, T. G., et al. (2009). *Basics of oncology*. Springer.

[Surma and Vanhoof, 1995] Surma, J. and Vanhoof, K. (1995). Integrating rules and cases for the classification task. In *International Conference on Case-Based Reasoning*, pages 325–334. Springer.

[Surma and Vanhoof, 1998] Surma, J. and Vanhoof, K. (1998). An empirical study on combining instance-based and rule-based classifiers. In *Proceedings of the AAAI '98 Spring Symposium on Multimodal Reasoning, AAAI Press, Stanford*.

[Sycara, 1990] Sycara, K. P. (1990). Persuasive argumentation in negotiation. *Theory and decision*, 28(3):203–242.

[Szarvas et al., 2007] Szarvas, G., Farkas, R., and Busa-Fekete, R. (2007). State-of-the-art anonymization of medical records using an iterative machine learning framework. *Journal of the American Medical Informatics Association*, 14(5):574–580.

[Tyczynski et al., 2003] Tyczynski, J. E., Démaret, D., and Parkin, D. M. (2003). *Standards and guidelines for cancer registration in Europe: the ENCR recommendations*. International Agency for Research on Cancer.

[Van Lent et al., 2004] Van Lent, M., Fisher, W., and Mancuso, M. (2004). An explainable artificial intelligence system for small-unit tactical behavior. In *Proceedings of the national conference on artificial intelligence*, pages 900–907. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999.

[Vellido, 2019] Vellido, A. (2019). The importance of interpretability and visualization in machine learning for applications in medicine and health care. *Neural Computing and Applications*, pages 1–15.

[Wang et al., 2010] Wang, D., Li, T., Zhu, S., and Gong, Y. (2010). ihelp: An intelligent online helpdesk system. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 41(1):173–182.

[World Health Organisation, 2013] World Health Organisation (2013). *International classification of diseases for oncology (ICD-O) – 3rd edition*.

[World Health Organization, 2019] World Health Organization (2019). Top 10 Causes of Death `https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death`, *Last consultation: November 2019*.

[World Wide Web Consortium, 1997] World Wide Web Consortium (1997). World wide web consortium publishes public draft of resource description framework.